

UNIVERSIDAD DE ALMERÍA

ESCUELA SUPERIOR DE INGENIERÍA

Monitorizar impresión de documentos en
Windows

Curso 2015/2016

Alumno/a:
Miguel Milla Arregui

Director/es:
Antonio Becerra Terón



INDICE

INDICE	3
1. Introducción	19
1.1 Descripción Sistema Actual: MegaTrack.....	20
1.2 Reemplazar SW Megatrack.....	22
1.3 Objetivo del TFG.....	24
1.4 Planificación.....	25
1.4.1 Fase 1: Viabilidad. Prototipos.....	25
1.4.2 Fase 2: Implementación. Modelo Incremental.....	25
1.4.3 Fase 3: Implantación. Scrum	26
1.4.4 Diagrama de Gantt.....	26
2. Estado del Arte	29
2.1 Arquitectura del Servidor de Impresión a partir de Windows 2000	30
2.1.1 Proveedor de impresión Local.....	32
2.1.2 Proveedor de Impresión Remoto	33
2.1.3 Funciones definidas por los Proveedores de Impresión	34
2.1.4 Ampliación de la arquitectura de impresión a partir de Windows Vista: doble ruta. 35	
2.2 Obtención de información de los trabajos de impresión.	39
2.2.1 Eventos del Servidor de Impresión	39
2.2.2 APIs proporcionadas por el Servidor de impresión	40
2.2.3 Estructuras utilizadas en las funciones descritas:.....	42
2.2.4 Namespace System.Printing de .NET	44
2.2.5 Análisis de ficheros de Spool del Sistema de impresión de Windows.	45
2.2.5.1 PJI (Printer Job Language).....	46
2.2.5.2 PCLXL	49
2.2.5.3 EMF y EMFSPOOL	59
2.2.5.4 Ejemplo de análisis de ficheros de spool.....	74
3. Metodología e infraestructura utilizada para el desarrollo del TFG	103
3.1 Equipos y dispositivos de impresión.....	103

3.2	Herramientas.....	104
3.2.1	Desarrollo SW y BD.	104
3.2.2	Utilidades.....	104
3.2.3	Planificación.....	105
3.2.4	BD	105
3.2.5	Ofimáticas.....	105
3.2.6	Servidor de aplicaciones	105
3.3	Metodologías	106
3.3.1	Modelo de Prototipos	106
3.3.2	Modelo Incremental.....	106
3.3.3	Scrum	107
3.3.3.1	¿Por qué utilizar una metodología ágil?.....	107
3.3.3.2	Scrum	109
4.	Desarrollo del TFG	117
4.1	Análisis del sistema.....	118
4.1.1	Identificación de usuarios y componentes con los que interactúa el Servicio Monitor de impresión.....	118
4.1.2	Requisitos SW.....	118
4.1.2.1	Requisitos Funcionales	118
4.1.2.2	Requisitos no funcionales	120
4.1.3	Casos de Uso.....	122
4.1.3.1	Diagrama de casos de uso.....	124
4.1.3.2	Especificación de casos de uso	124
4.2	Diseño y codificación.....	128
4.2.1	Lógica del Servicio Monitor Impresión.....	128
4.2.2	Diagrama de clases	129
4.2.3	BD	131
4.2.4	Enumeraciones y estructuras empleadas para implementar las clases.....	133
4.2.4.1	Estructuras.....	133
4.2.4.2	Enumeraciones.....	137
4.3	Implementación. Clases	139
4.3.1	Clase EvenMonitorPrinting.....	139
4.3.1.1	Documentación del constructor.....	139
4.3.1.2	Documentación de las funciones miembro	139
4.3.1.3	Documentación de los datos miembro.....	142

4.3.2	Clase AnalyzeJobId	144
4.3.2.1	Documentación del constructor.....	144
4.3.2.2	Documentación de las funciones miembro	145
4.3.2.3	Documentación de los datos miembro.....	148
4.3.3	Clase LocalPrinting.....	148
4.3.3.1	Documentación del constructor.....	149
4.3.3.2	Documentación de las funciones miembro	149
4.3.3.3	Documentación de los datos miembro.....	151
4.3.4	Clase ApiImpresion	151
4.3.4.1	Documentación del constructor.....	151
4.3.4.2	Documentación de las funciones miembro	151
4.3.4.3	Documentación de los datos miembro.....	158
4.3.5	Clase Spool	158
4.3.5.1	Documentación del constructor.....	159
4.3.5.2	Documentación de las funciones miembro	159
4.3.5.3	Documentación de los datos miembro.....	160
4.3.6	Clase EmfSpool.....	160
4.3.6.1	Documentación del constructor.....	160
4.3.6.2	Documentación de las funciones miembro	161
4.3.6.3	Documentación de los datos miembro.....	163
4.3.7	Clase JPLSpool	163
4.3.7.1	Documentación del constructor.....	163
4.3.7.2	Documentación de las funciones miembro	164
4.3.7.3	Documentación de los datos miembro.....	166
4.3.8	Clase PCLXLSpool.....	167
4.3.8.1	Documentación del constructor.....	167
4.3.8.2	Documentación de las funciones miembro	168
4.3.8.3	Documentación de los datos miembro.....	169
4.3.9	Clase WatchIO.....	170
4.3.9.1	Documentación del constructor.....	170
4.3.9.2	Documentación de las funciones miembro	170
4.3.9.3	Documentación de los datos miembro.....	172
4.3.10	Clase Programa	172
4.3.10.1	Documentación del constructor	172
4.3.10.2	Documentación de las funciones miembro.....	173

4.3.10.3	Documentación de los datos miembro.....	173
4.3.11	Clase ServicePrintMonitor	173
4.3.11.1	Documentación del constructor.....	174
4.3.11.2	Documentación de las funciones miembro.....	174
4.3.11.3	Documentación de los datos miembro.....	174
4.3.12	Interface IDB.....	175
4.3.13	Clase DB	176
4.3.13.1	Documentación del constructor.....	176
4.3.13.2	Documentación de las funciones miembro.....	176
4.3.13.3	Documentación de los datos miembro.....	176
4.3.14	Clase DBOracle	177
4.3.14.1	Documentación del constructor y destructor.....	178
4.3.14.2	Documentación de las funciones miembro.....	178
4.3.14.3	Documentación de los datos miembro.....	179
4.4	Implantación con SCRUM	180
4.4.1	Reunión de planificación del proyecto.....	181
4.4.1.1	Duración de las iteraciones.....	181
4.4.1.2	Velocidad de trabajo (VT).....	181
4.4.1.3	Delimitar el ámbito de la implantación.....	181
4.4.1.4	Funcionalidades (Product Backlog).	182
4.4.2	Primera iteración	185
4.4.2.1	Sprint planning Meeting (Reunión de la iteración): Objetivos.....	185
4.4.2.2	Desarrollo iteración	187
4.4.2.3	Sprint Review (Reunión demostración).....	194
4.4.2.4	Sprint retrospective	196
4.4.3	Segunda Iteración	198
4.4.3.1	Sprint planning Meeting (Reunión de la iteración): Objetivos.....	198
4.4.3.2	Desarrollo iteración	199
4.4.3.3	Sprint Review (Reunión demostración).....	206
4.4.3.4	Sprint retrospective	208
4.4.4	Tercera iteración.....	210
4.4.4.1	Sprint planning Meeting (Reunión de la iteración): Objetivos.....	210
4.4.4.2	Desarrollo iteración	210
4.4.4.3	Sprint Review (Reunión demostración).....	212
4.4.4.4	Sprint retrospective	213

4.5	Valoración Económica	214
4.5.1	Recursos humanos.....	214
4.5.2	Costo por recursos materiales.....	216
5.	Conclusiones y trabajo futuro	219
6.	Bibliografía.....	221
7.	Anexos.....	223
7.1	Creación de un Servicio WCF para la capa CAD.	223
7.2	Cliente Oracle con ODP.NET.....	225
7.3	Log4net.....	226
7.4	Referencia de archivos del proyecto con código fuente.	229
7.4.1	Servicio Monitor Impresión.....	229
7.4.1.1	Archivo .\ServicioMonitorImpresion\MonitorImpresion\AnalyzeJobId.cs	229
7.4.1.2	Archivo .\ServicioMonitorImpresion\MonitorImpresion\ApiImpresion.cs	229
7.4.1.3	Archivo .\ServicioMonitorImpresion\MonitorImpresion\EmfSpool.cs.....	230
7.4.1.4	Archivo .\ServicioMonitorImpresion\MonitorImpresion\EnumDictionaryStruct.cs.....	230
7.4.1.5	Archivo .\ServicioMonitorImpresion\MonitorImpresion\EventMonitorPrinting.cs.....	231
7.4.1.6	Archivo .\ServicioMonitorImpresion\MonitorImpresion\JPLSpool.cs	231
7.4.1.7	Archivo .\ServicioMonitorImpresion\MonitorImpresion\LocalPrinting.cs.....	231
7.4.1.8	Archivo .\ServicioMonitorImpresion\MonitorImpresion\PCLXLSpool.cs.....	231
7.4.1.9	Archivo .\ServicioMonitorImpresion\MonitorImpresion\Program.cs.....	232
7.4.1.10	Archivo .\ServicioMonitorImpresion\MonitorImpresion\ProjectInstaller.cs 232	
7.4.1.11	Archivo .\ServicioMonitorImpresion\MonitorImpresion\ProjectInstaller.Designer.cs.....	232
7.4.1.12	Archivo .\ServicioMonitorImpresion\MonitorImpresion\ServicePrintMonitor.cs.....	232
7.4.1.13	Archivo .\ServicioMonitorImpresion\MonitorImpresion\Spool.cs	232
7.4.1.14	Archivo .\ServicioMonitorImpresion\MonitorImpresion\Util.cs	233
7.4.1.15	Archivo .\ServicioMonitorImpresion\MonitorImpresion\WatchIO.cs.....	233
7.4.1.16	Archivo .\ServicioMonitorImpresion\MonitorImpresionTest\Test.cs.....	233
7.4.2	Servicio WCF para implementar la capa CAD.....	234
7.4.2.1	Archivo .\WCFDBOracle\WCFDBOracle\DB.cs.....	234
7.4.2.2	Archivo .\WCFDBOracle\WCFDBOracle\DBOracle.cs	234
7.4.2.3	Archivo .\WCFDBOracle\WCFDBOracle\IDB.cs	234

8.	Glosario de términos.....	235
-----------	---------------------------	-----

Índice de ilustraciones

Ilustración 1 Arquitectura de componentes SW Megatrack.	21
Ilustración 2 Arquitectura del sistema propuesto.	22
Ilustración 3. Diagrama de Gantt	27
Ilustración 4 Sistema Impresión Windows	30
Ilustración 5 Sistema Impresión Windows. Detalle proveedor impresión local	32
Ilustración 6 Sistema Impresión Windows. Detalle proveedor impresión remoto	34
Ilustración 7 Arquitectura Sistema Impresión: doble ruta	36
Ilustración 8 Arquitectura Sistema Impresión: doble ruta. Módulos conversores	37
Ilustración 9 Ejemplo de Stream Header.	54
Ilustración 10 Imagen obtenida del documento MSDN: [MS-EMF].pdf	60
Ilustración 11 Imagen obtenida del documento MSDN: [MS-EMFSPool].pdf	68
Ilustración 12 Ejemplo para análisis de fichero de spool. Contenido primera página	74
Ilustración 13 Ejemplo para análisis de fichero de spool. Contenido segunda página	74
Ilustración 14. Modelo Prototipo. Fuente (Inteco, 2009)	106
Ilustración 15. Modelo Incremental. Fuente (Inteco, 2009)	107
Ilustración 16 Ejemplo Diagrama Burndown.	111
Ilustración 17 Modelo Scrum. Fuente: http://queondara.readthedocs.org/en/latest/Scrum.html	114
Ilustración 18. Diagrama caso de uso: Actor.	122
Ilustración 19. Diagrama caso de uso: Caso de uso.	122
Ilustración 20. Diagrama caso de uso: Relación de asociación.	122
Ilustración 21. Diagrama caso de uso: Relación de dependencia.	122
Ilustración 22. Diagrama caso de uso: Relación incluye.	123
Ilustración 23. Diagrama caso de uso: Relación extiende.	123
Ilustración 24. Diagrama casos de uso del Servicio Monitor Impresión.	124
Ilustración 25. Flujograma. Captura propiedades de trabajo de impresión.	128
Ilustración 26. Diagrama de clases de Servicio Monitor Impresión	129
Ilustración 27. Diagrama de clases del Servicio WCF para la capa CAD.	130
Ilustración 28. Tabla Oracle SPOOL_PRINTJOBS para almacenar los trabajos de impresión	132
Ilustración 29. Clase EventMonitorPrinting.	139
Ilustración 30. Clase AnalyzeJobId	144
Ilustración 31. Gráfico de llamadas desencadenadas por el método AnalyzePrintJob.	146
Ilustración 32. Clase LocalPrinting.	148
Ilustración 33. Propiedad de impresora para conservar documentos después de imprimir.	149
Ilustración 34. Clase ApiImpresión.	151
Ilustración 35. Gráfico de llamadas para el método ApiPrintJob.Analyze.	152
Ilustración 36. Gráfico de llamadas de la función GetPrinter5.	155
Ilustración 37. Gráfico de llamadas para la función KeepSpoolFiles.	156
Ilustración 38. Clase Spool.	158
Ilustración 39. Diagrama de herencia sobre la clase Spool.	159
Ilustración 40. Clase EmfSpool.	160
Ilustración 41. Clase JPLSpool.	163
Ilustración 42. Gráfico de llamadas del método JPLSpool.Analyze	165
Ilustración 43. Clase PCLXLSpool.	167
Ilustración 44. Clase WatchIO.	170
Ilustración 45. Clase Programa.	172
Ilustración 46. Clase ServicePrintMonitor.	173
Ilustración 47. Gráfico de llamadas al constructor de la clase ServicePrintMonitor.	174
Ilustración 48. Diagrama de colaboración de la clase DBOracle.	178
Ilustración 49. Diagrama de componentes del sistema a implantar.	182
Ilustración 50. Gráfica burndown de la 1ª iteración.	187

<i>Ilustración 51. Tareas de E.T 2.4</i>	187
<i>Ilustración 52. Instalación del Servicio Monitor de Impresión.</i>	188
<i>Ilustración 53. Servicio Monitor de Impresión.</i>	189
<i>Ilustración 54. Tareas de E.T 2.1</i>	189
<i>Ilustración 55. Formulario Access para consultar y hacer anotaciones en trabajos de impresión.</i>	190
<i>Ilustración 56. Tareas de E.T 1.1</i>	191
<i>Ilustración 57. Captura de resultados de las pruebas automatizadas.</i>	192
<i>Ilustración 58. Tareas de E.T 1.2</i>	193
<i>Ilustración 59. Tareas de E.T 1.3</i>	193
<i>Ilustración 60. Tareas de E.T 2.3</i>	194
<i>Ilustración 61. Funcionalidad F.10</i>	199
<i>Ilustración 62. Gráfica burndown de la 2ª iteración.</i>	200
<i>Ilustración 63. Tareas de E.T 2.5</i>	200
<i>Ilustración 64. Hoja de registro de trabajos de impresión.</i>	201
<i>Ilustración 65. Ejemplos de notificaciones por email de impresión en curso.</i>	202
<i>Ilustración 66. Tareas de E.T 2.6</i>	203
<i>Ilustración 67. Tareas de E.T 2.7</i>	205
<i>Ilustración 68. . Tareas de E.T 10.1</i>	206
<i>Ilustración 69. Gráfica burndown de la 3ª iteración</i>	211
<i>Ilustración 70. Tareas de E.T 2.8</i>	211
<i>Ilustración 71. Tareas de E.T 10.2</i>	212
<i>Ilustración 72. Recursos humanos asignados a tareas.</i>	215
<i>Ilustración 73. Agregar referencia a componente ODP.NET de Oracle.</i>	225

Índice de tablas

Tabla 1. Eventos de impresión	40
Tabla 2. Sintaxis instrucción JPL	46
Tabla 3. PCLXL. Atributos de estado gráfico	52
Tabla 4. Cabecera PCLXL	53
Tabla 5. Cabecera PCLXL. Identificador de formato.	54
Tabla 6. Plantilla requisito	118
Tabla 7. Requisito RF-1	118
Tabla 8. Requisito RF-2	118
Tabla 9. Requisito RF-3	119
Tabla 10. Requisito RF-4	119
Tabla 11. Requisito RF-5	119
Tabla 12. Requisito RF-6	119
Tabla 13. Requisito RF-7	119
Tabla 14. Requisito RF-8	120
Tabla 15. Requisito RNF-1	120
Tabla 16. Requisito RNF-2	120
Tabla 17. Requisito RNF-3	120
Tabla 18. Requisito RNF-4	120
Tabla 19. Requisito RNF-5	121
Tabla 20. Requisito RNF-6	121
Tabla 21. Requisito RNF-7	121
Tabla 22. Requisito RNF-8	121
Tabla 23. Ficha para describir casos de uso.	123
Tabla 24. Especificación caso de uso. CU1	125
Tabla 25. Especificación caso de uso. CU2	125
Tabla 26. Especificación caso de uso. CU3	125
Tabla 27. Especificación caso de uso. CU4	126
Tabla 28. Especificación caso de uso. CU5	126
Tabla 29. Especificación caso de uso. CU6	126
Tabla 30. Especificación caso de uso. CU7	127
Tabla 31. Especificación caso de uso. CU8	127
Tabla 32. Especificación caso de uso. CU9	127
Tabla 33. Breve descripción de las clases más importantes.	131
Tabla 34. Ejemplo código C# suscripción a Eventos de Impresión	142
Tabla 35. Scrum. Product Backlog	184
Tabla 36. E.T asociados a funcionalidad F.1	186
Tabla 37. E.T asociados a funcionalidad F.2	186
Tabla 38. Nuevos campos para tabla SPOOL_PRINTJOBS para plotter y tipo de papel.	194
Tabla 39. Total impresiones por plotter primera iteración..	195
Tabla 40. Total impresiones por plotter según número de copias primera iteración.	195
Tabla 41. Impresiones por día.	195
Tabla 42. Impresiones por tamaño de papel.	196
Tabla 43. Detalles trabajos de impresión iteración 1.	197
Tabla 44. nuevos E.T asociados a funcionalidad F.2	198
Tabla 45. . E.T asociados a funcionalidad F.10	199
Tabla 46. Total impresiones por plotter segunda iteración.	206
Tabla 47. Total impresiones por plotter según número de copias en segunda iteración.	206
Tabla 48. Número de impresiones por día segunda iteración.	207
Tabla 49. Impresiones por tamaño de papel segunda iteración.	207
Tabla 50. nuevo E.T asociado a funcionalidad F.2	210

<i>Tabla 51. nuevo E.T asociado a funcionalidad F.10</i>	<i>210</i>
<i>Tabla 52. Coste total del TFG.</i>	<i>214</i>
<i>Tabla 53. Coste de recurso humano por hora.</i>	<i>214</i>
<i>Tabla 54. Coste total por recurso humano.</i>	<i>216</i>
<i>Tabla 55. Costo por recursos materiales.</i>	<i>217</i>

Índice de detalles de codificación, estructuras de datos y resultados

Detalle 1. Estructura JOB_INFO_2	43
Detalle 2. Estructura DEVMODE	43
Detalle 3. Estructura PRINTER_DEFAULTS	44
Detalle 4. Estructura PRINTER_INFO5	44
Detalle 5. Enumeración con operadores PCLXL	51
Detalle 6. Enumeración con atributos PCLXL	57
Detalle 7. Enumeración con tipos de datos PCLXL	57
Detalle 8. Enumeración EMR_RECORD. tipos de registros EMF	63
Detalle 9. Registros EMF. Mapas de bits	63
Detalle 10. Registros EMF. Definición de regiones	64
Detalle 11. Registros EMF. Comentarios	64
Detalle 12. Registros EMF. Control	64
Detalle 13. Registros EMF. Dibujar	65
Detalle 14. Registros EMF. Secuencia de escape	65
Detalle 15. Registros EMF. Creación de objetos	66
Detalle 16. Registros EMF. Manipulación de objetos	66
Detalle 17. Registros EMF. OpenGL	66
Detalle 18. Registros EMF. Rutas de dibujo	66
Detalle 19. Registros EMF. Estado	67
Detalle 20. Registros EMF. Transformación	67
Detalle 21. Registros EMFSPPOOL.	70
Detalle 22. Registros EMFSPPOOL. Con contenido de página.	70
Detalle 23. Registros EMFSPPOOL. De posición de página.	71
Detalle 24. Registros EMFSPPOOL. Definición de fuentes.	71
Detalle 25. Registros EMFSPPOOL. Posición de fuentes.	72
Detalle 26. Registros EMFSPPOOL. _DEVMODE.	72
Detalle 27. Estructura STRUCT_PRINT_JOB para guardar los detalles de los trabajos de impresión.	134
Detalle 28. Estructura JOB_INFO_2. Utilizada con la API GetJob	135
Detalle 29. Estructura SYSTEMTIME. Para guardar una fecha y hora	135
Detalle 30. Estructura PRINTER_INFO_5	136
Detalle 31. Estructura PRINTER_DEFAULTS	136
Detalle 32. Estructura DEVMODE	137
Detalle 33. Enumeración EVENTS	138
Detalle 34. API sistema impresión. ClosePrinter	152
Detalle 35. API sistema impresión. GetJob	152
Detalle 36. API sistema impresión. Uso GetJob	153
Detalle 37. API sistema impresión. GetPrinter	154
Detalle 38. API sistema impresión. OpenPrinter	156
Detalle 39. API sistema impresión. SetPrinter	157
Detalle 40. Clase WatchIO. declaración método FileEvent	170
Detalle 41. Interfaz de servicio WCF	175
Detalle 42. Ejemplo de codificación de prueba automatizada.	192
Detalle 43. Ejemplo de resumen de tags en análisis EMF	204
Detalle 44. Análisis EMF. log ejemplo con obtención de tamaño de página	205
Detalle 45. Error análisis fichero spool EMF. Fichero en uso	208
Detalle 46. T 2.8.2 lógica de prueba copia ficheros grandes	212

Palabras clave

EMF, SPOOL, GDI, JPL, Log4Net, PCLXL, PDL, Plotter, Printing, Spooler, Scrum, System.Printing, Winspool.drv.

RESUMEN

En este documento se explica el funcionamiento del sistema de impresión de Windows y su interacción con otros componentes del sistema operativo. El Sistema de impresión de Windows registra su actividad en el registro de eventos de Windows (eventos de impresión) y almacena los trabajos de impresión en ficheros de spool empleando alguno de los lenguajes de descripción de páginas (PDL).

El objetivo de este TFG es poder obtener las propiedades de los trabajos de impresión con las que poder calcular sus costes de impresión.

A través de los eventos de impresión es posible identificar los trabajos de impresión (Número de trabajo de impresión, impresora, Servidor de impresión, usuario propietario). Mediante el namespace System.Printing de .NET y las APIs del sistema de impresión se obtienen propiedades como: nombre del documento, número de páginas, impresión en color o b/n, impresión simple o doble cara, tamaño de página, fecha de impresión. Analizando los ficheros de spool (empleando lenguajes PDL: EMFSPool/EMF o JPL/PCLXL), es posible obtener ciertas propiedades como el número de páginas con contenido en color o el tamaño de cada una de las páginas del documento.

Se utiliza Visual Studio (con las extensiones log4net y ODP.NET) y C# para crear un Servicio Windows con el que monitorizar los trabajos de impresión y un servicio WCF para crear la capa de acceso a datos para almacenar las propiedades de los trabajos de impresión en una BD Oracle.

Destacar que surge como necesidad utilizar un marco de trabajo como SCRUM, junto Visual Studio TFS, para concluir la última fase de este TFG.

ABSTRACT

This document shows the operation of Windows printing system and its interaction with other components of the operating system. The Windows printing system allows to record its activity in the Windows event log (printing events) and it stores printjobs in the spool files by using any of the page description languages (PDL).

The aim of this project is to obtain the properties of print jobs in order to calculate their printing costs.

The identification of printing jobs is possible by means of the printing events (such as number of print job, printer, print server, owner user). Different features, like document name, number of pages, print in color or b/n, simple or double-sided printing, page size, print date, among others, can be obtained using the .NET namespace System.Printing and the API printing system. Moreover, analyzing spool files (using PDL languages as EMFSPool/EMF or JPL/PCLXL), it is possible to obtain certain properties such as number of pages with color content or the size of each page of the document.

Finally, let us remark that Visual Studio (with log4net and Oracle ODP.NET extensions) and C# are used in order to create a Windows and WCF services, which allows to monitor printjobs and to create a data access layer in order to save the properties of printjobs in an Oracle database, respectively.

In order to complete the last stage of this project, it is important to remark the use of a developer framework such as Scrum with Visual Studio Team Foundation Server.

1. ■ INTRODUCCIÓN

Durante los días en los que andaba buscando alguna idea para poder hacer mi TFG coincidió que uno de los servicios ofrecidos desde el STIC de la UAL, donde trabajo, quedo fuera de servicio.

Se trataba del servicio de impresión ofrecido desde el STIC destinado a PDI y PAS. Este servicio está instalado en un equipo HP antiguo, con S.O Windows 2003 en el que tenemos instalado el SW Megatrack, de la empresa Bluemega, para monitorizar los trabajos de impresión y poder calcular los costes que generan. El equipo HP se averió.

¿Cómo solucionarlo? La primera idea sustituir el ya decrépito equipo por uno nuevo y con mejores prestaciones en el que se debería hacer una nueva instalación del SW Megatrack, pero nos encontramos con las siguientes restricciones del SW Megatrack con el que se monitoriza y se registran los trabajos de impresión:

- La empresa Bluemega propietaria del SW Megatrack ha desaparecido.
- Como consecuencia no tenemos soporte para el SW Megatrack
- La instalación del SW Megatrack requiere de la generación de una clave de instalación facilitada por la empresa Bluemega para el equipo específico donde se va a instalar.

Por estas razones la única solución para seguir utilizando el SW Megatrack pasaba por intentar reparar el equipo averiado. Tras revisar el equipo se ve que lo que falla es la placa base del mismo; por suerte disponíamos de un equipo idéntico, por lo que tras cambiar la placa base el servidor vuelve a funcionar con normalidad.

De aquí surge la idea de desarrollar un SW para poder monitorizar los trabajos de impresión en Sistemas Operativos Windows, con el que poder capturar los detalles de cada trabajo de impresión y guardarlos en una BD. Llamaremos a este SW Monitor de Impresión.

Con los datos capturados con el Monitor de Impresión es posible desarrollar una aplicación con la que calcular costes de impresión.

1.1 Descripción Sistema Actual: MegaTrack

Estas son sus características:

- Para utilizar el servicio el usuario tiene que identificarse con una cuenta del dominio de red Microsoft en la red de la UAL (dominio WUAL).
- Para cada trabajo de impresión se guarda la siguiente información:
 - ✓ Usuario (se corresponde con la cuenta de usuario en el dominio WUAL)
 - ✓ Impresora
 - ✓ Número total de páginas impresas
 - ✓ Número de páginas impresas en color
 - ✓ Tamaño de papel
 - ✓ Fecha de impresión
 - ✓ Impresión simple o a doble cara.
 - ✓ Grupo del dominio WUAL al que pertenece el usuario
- Mensualmente se genera un listado (Excel) con los costes de impresión de cada usuario. Este listado Excel tiene que ser revisado, en un proceso externo al SW Megatrack; para ello se carga en una BD Access para poder cruzar con las B.D corporativas (cruzando datos a través de la cuenta de dominio WUAL con la que se lanzó el trabajo de impresión) y así poder obtener además la Unidad o Departamento a la que pertenece el usuario y su Nombre y Apellidos. A partir de este listado Excel se genera un documento resumen con gastos de impresión enviado al Servicio de Asuntos Económicos de la UAL a través de registro electrónico y a las distintas unidades con centro de gasto a través de una aplicación de notificaciones.
- El SW Megatrack solamente está instalado en un servidor y sólo se monitorizan las impresoras instaladas en este servidor.
- El SW Megatrack compuesto por un proceso de monitorización de trabajos de impresión, un frontal para poder gestionar los trabajos de impresión y una BD donde se almacenan las propiedades de los trabajos de impresión residen en el mismo servidor.

Carencias:

- Microsoft dejó de dar soporte oficialmente a Windows 2003 el 14 de julio de 2015.
- El SW Megatrack no permite monitorizar los trabajos de impresión de los modelos de plotter recientes ofrecidos dentro de los servicios de impresión del STIC.
- No permite una arquitectura de varios niveles (p.e. Capa de negocio, frontal, Capa de acceso a datos y SGBD).
- Generación manual del resumen de información enviada al Servicio de Asuntos Económicos y a las Unidades con centro de gasto asociado a las que se les va a cargar los costes de impresión a partir del fichero Excel generado por la aplicación Megatrack.

En la siguiente ilustración se muestra un esquema del sistema actual:

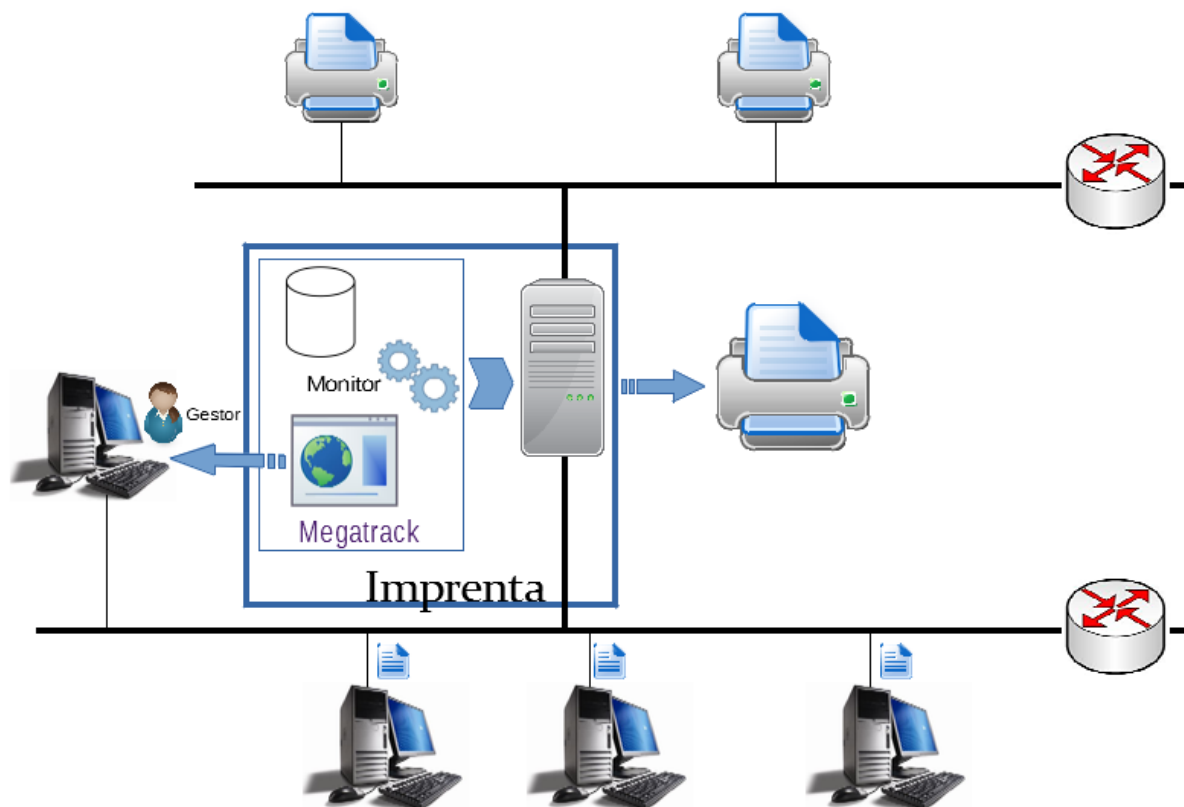


Ilustración 1 Arquitectura de componentes SW Megatrack.

1.2 Reemplazar SW Megatrack.

Propongo, a los responsables del Área de Sistemas del STIC, desarrollar un SW propio para reemplazar Megatrack y suplir las carencias puestas de manifiesto en el punto anterior, cubriendo las funcionalidades existentes y añadiendo las siguientes mejoras:

1. Evitar los inconvenientes de un SW propietario:
 - a. Restricciones de licencia (instalaciones permitidas, renovaciones periódicas, ampliación funcionalidades).
 - b. Desaparición de la empresa propietaria del SW; con lo que desaparece el soporte ante posibles fallos de la aplicación e inevitables actualizaciones del entorno de ejecución de la aplicación: Sistema Operativo, nuevo HW.
2. Mejorar la modularidad de la aplicación para poder alojar cada uno de los módulos de la aplicación en el entorno más adecuado; todos los componentes de Megatrack se alojan en el mismo servidor (Módulo monitor de trabajos de impresión, frontal gestión trabajos de impresión, SGBD). De este modo se puede aplicar el nivel de seguridad que estimemos más adecuado para cada módulo.

En la siguiente ilustración se muestra la arquitectura en varias capas:

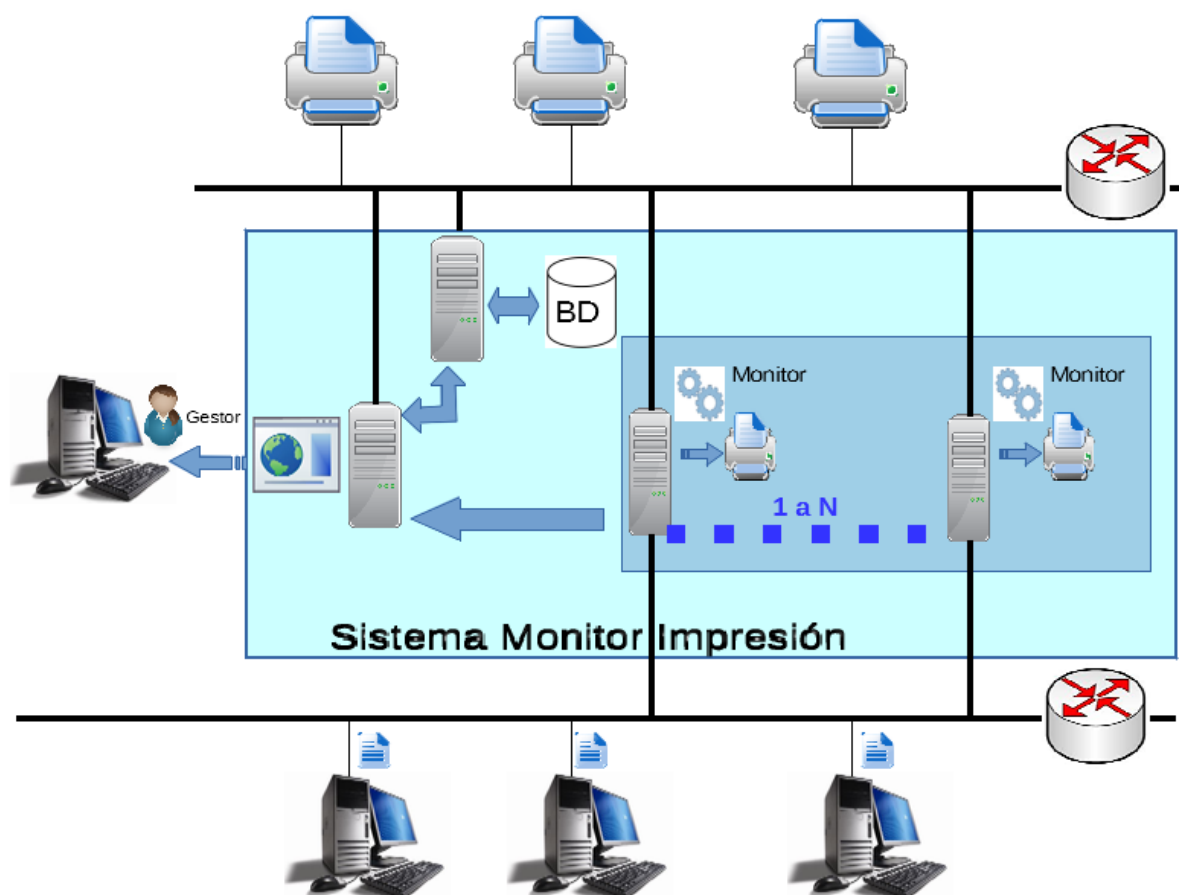


Ilustración 2 Arquitectura del sistema propuesto.

La idea es dividir la aplicación en 4 capas principales:

Monitor de Impresión: Instalado, como servicio de Windows, en cualquier servidor que tenga instalado un dispositivo de impresión local o en red, en el que se quiere monitorizar los trabajos de impresión que lanzan los usuarios y poder obtener sus características.

CAD: Esta capa aísla el acceso a la BD. Va a consistir en un Servicio Web que proporcionará la funcionalidad necesaria para interactuar con el S.G.B.D.

S.G.B.D (Oracle): Donde se implementa la BD para almacenar las propiedades de los trabajos de impresión capturados con el Monitor de Impresión.

Frontal de Gestión: Aplicación para explotar la información capturada por el Monitor de Impresión. El usuario gestor puede obtener la información recogida de los trabajos de impresión, generando listados mensuales con los costes de impresión, como valor principal.

1.3 Objetivo del TFG.

El objetivo es poder obtener los detalles de los trabajos de impresión, lanzados por los usuarios, que en un proceso posterior permita calcular los costes de impresión generados por cada usuario.

Es decir, de las 4 capas de la arquitectura presentadas en el punto anterior, este TFG se va a centrar en la implementación de la capa Monitor de Impresión que es con la que se obtiene los detalles de los trabajos de impresión. Además, bajo mi punto de vista, es la que entraña mayor dificultad y es su viabilidad la que condiciona la realización del resto de capas. También se implementará parcialmente la capa CAD y una BD muy simple (en principio una única tabla) para guardar la información capturada de los trabajos de impresión. Para ello:

- Se estudiará el sistema de impresión Windows.
- Del estudio se debe descubrir de qué fuentes se puede obtener información de los trabajos de impresión.
- Identificadas las posibles fuentes para obtención de información de los trabajos de impresión, se implementará el servicio monitor de impresión, comprobando que se obtienen los detalles necesarios para poder calcular los costes de impresión. Es fundamental obtener los detalles de impresión de los plotters, que no están soportados en Megatrack.

Un objetivo adicional es la adquisición de los conocimientos necesarios en el uso de las herramientas, métodos y lenguajes utilizados para realizar el TFG:

- ✓ Lenguaje programación: C#.
- ✓ Desarrollo Web Services: WCF.
- ✓ Creación Servicios de sistema.
- ✓ Familiarización con el Framework .NET de Microsoft.
- ✓ Visual Studio como IDE de desarrollo que integra y facilita el trabajo.
- ✓ Visual Studio Tema Foundation Server

Se deja como trabajo posterior a este TFG la implementación del frontal de Gestión junto a la implementación final de la BD y la capa CAD.

1.4 Planificación

Para la consecución del este TFG se han considerado 3 fases. La primera fase, condición necesaria, para abordar con garantía la realización del TFG seleccionado o descartarlo.

1.4.1 Fase 1: Viabilidad. Prototipos.

Seleccionado el tema que quiero abordar con mi TFG, monitorizar trabajos de impresión para poder aplicarlo en el calculo de costes de impresión, en una primera fase hay una cuestión fundamental, ¿Es viable? (fundamentalmente en la parte técnica. De la parte económica y temporal en principio se dispone de medios suficientes), ya que en el momento en el que decido abordar este trabajo para el TFG tengo una idea muy vaga de como funciona el Sistema de impresión de Windows y las dificultades que entraña poder interactuar con él.

En primer lugar hay que realizar una importante labor de investigación para identificar los posibles métodos para la obtención de información de los detalles de los trabajos de impresión. Este trabajo queda reflejado en el Título 2 del TFG.

En segundo lugar hay que seleccionar el entorno de desarrollo junto el lenguaje de programación que mejor se adapte y me resulte más cómodo para implementar la solución. Esto añade un extra de tiempo a emplear en formación, también una motivación extra, debido a que mi perfil laboral no es de desarrollador y no estoy familiarizado con las herramientas de desarrollo y lenguajes actuales. En el título 3 se detallan las herramientas utilizadas.

Finalmente en esta primera fase, una vez tenemos el conocimiento y las herramientas, se trata de probar que es posible monitorizar los trabajos de impresión y obtener los detalles de impresión necesarios para poder calcular costes de impresión. Para ello se hacen implementaciones lo más simples posibles, mediante prototipos, para interactuar con el sistema de impresión, utilizando los métodos encontrados.

1.4.2 Fase 2: Implementación. Modelo Incremental.

Basándome en el trabajo hecho en la primera fase delimito el alcance del TFG que va a consistir en implementar un servicio Windows para monitorizar el servidor de impresión y obtener los detalles de los trabajos de impresión. Estos detalles se deben almacenar en una BD por lo que también hay que implementar una capa CAD (capa de acceso a datos); esta consiste en un servicio WCF instalado en un Servidor de aplicaciones IIS que aísla el acceso a la BD, alojada en un S.G.B.D Oracle.

En esta fase sigo un desarrollo incremental. Se han realizado los siguientes incrementos:

1. Eventos Impresión: Consiste en un Servicio de sistema que permite capturar los eventos del Sistema de impresión y generar un log con los detalles de las acciones realizadas.
2. API Sistema impresión: En este incremento se le añade las funcionalidades necesarias para capturar los detalles de los trabajos de impresión utilizando la API del sistema de impresión Windows y almacenarlos en una BD Oracle utilizando un servicio WCF que implementa la capa CAD.
3. Spool de Impresión: En este incremento se le añade la funcionalidad para obtener detalles de los trabajos de impresión analizando los ficheros de Spool.

1.4.3 Fase 3: Implantación. Scrum

Al finalizar la segunda fase ya disponemos de un producto funcional. Pero para su implantación considero necesario:

- Presentar el Sistema Monitor de Impresión a los usuarios que gestionan el Servicio de Impresión ofrecido desde STIC (usuarios de la futura aplicación para cálculo de costes de impresión de la que formará parte este Servicio Monitor de Impresión).
- Revisar los requisitos e incorporar nuevos requisitos si así lo piden los usuarios.
- Tener funcionando el Servicio Monitor de impresión durante un periodo de prueba en una situación de trabajo real, con los siguientes objetivos:
 - Capturar trabajos de impresión para que los operadores validen si los detalles de los trabajos de impresión capturados son correctos.
 - Detectar posibles errores de funcionamiento.
 - Comprobar que cumple con los requisitos. Si es necesario añadir nuevos requisitos.

Para ello necesitamos un marco de trabajo en el que los usuarios de la aplicación estén más involucrados y motivados. Por eso decido utilizar SCRUM para esta tercera fase del TFG.

1.4.4 Diagrama de Gantt

En la próxima ilustración presento el diagrama de Gantt asociado a este TFG.

Se puede observar que la dedicación del desarrollador ha ido variando en las distintas etapas del proyecto.

Incluyo de forma implícita la tarea “Documentar TFG” en las últimas etapas, cuando realizo una dedicación exclusiva, aunque realmente esta tarea se ha ido solapando al resto de tareas durante todo el TFG.

En la primera fase, el estudio (estado del arte), la formación (en las herramientas de desarrollo) y las pruebas realizadas para probar los posibles métodos de obtención de los detalles de los trabajos de impresión se han ido solapando en el tiempo.

Entre el final de la primera fase y la segunda hay un lapsus de tiempo en el que interrumpo el trabajo dedicado al TFG.

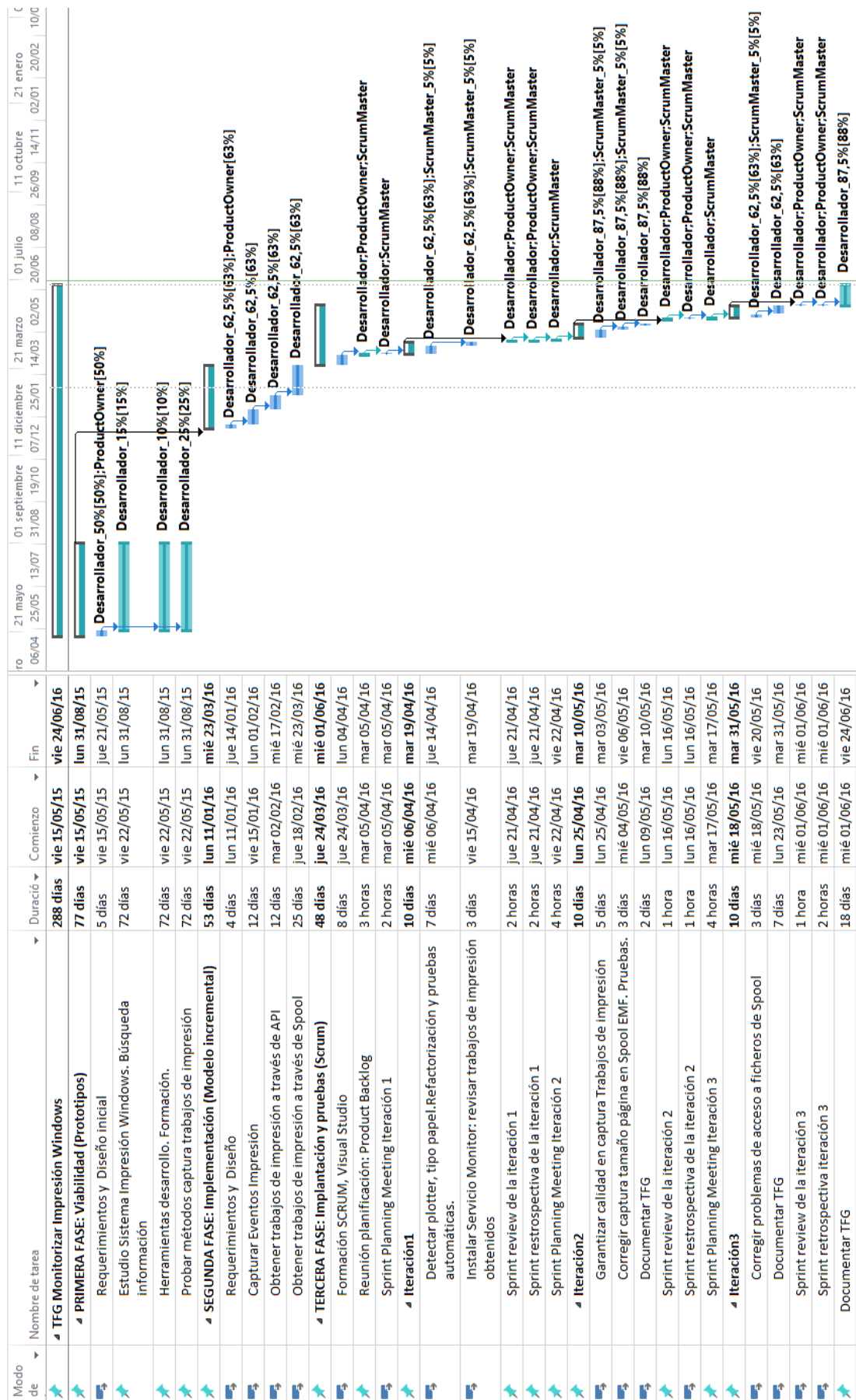


Ilustración 3. Diagrama de Gantt

2. ESTADO DEL ARTE

Como punto de partida no puedo dejar de hacer mención especial a la wiki dedicada a todo lo relacionado con la impresión (“Undocumented Printing Wiki [Undocumented Printing],” n.d.). En ella se puede encontrar mucho material no documentado oficialmente y referencias muy valiosas que me han ayudado a recorrer este camino.

Por supuesto es de lectura obligada el contenido del MSDN de Microsoft dedicado a la impresión en Windows (MSDN Microsoft, n.d.-c).

2.1 Arquitectura del Servidor de Impresión a partir de Windows 2000

La arquitectura del servidor de impresión a partir de Microsoft Windows 2000 y posteriores versiones de Sistemas Operativos de Microsoft consiste en un Gestor de *Colas de Impresión* (Print Spooler o Spooler) y un conjunto de drivers de Impresora. Las aplicaciones utilizando funciones independientes del dispositivo de impresión pueden crear trabajos de impresión y enviarlo a cualquier dispositivo.

Los controladores o drivers de la impresora incluye:

- Un DLL gráfico de impresión (**Printer Graphics DLL**) o componente de representación. Este componente se utiliza en conjunción con el GDI para la representación gráfica del trabajo de impresión enviando un flujo de datos a la cola de impresión en un formato que entiende la impresora.
- Un DLL de interfaz de impresora (**Printer Interface DLL**) o componente de configuración. Proporciona un interfaz de usuario para definir las características del trabajo de impresión como: tamaño de papel, número de copias, impresión color o b/n, impresión simple o a doble cara, etc. También proporciona un interfaz que el servidor de impresión puede utilizar para notificar al driver eventos relacionados con el sistema de impresión.

El siguiente diagrama muestra la arquitectura básica de los componentes del Spooler:

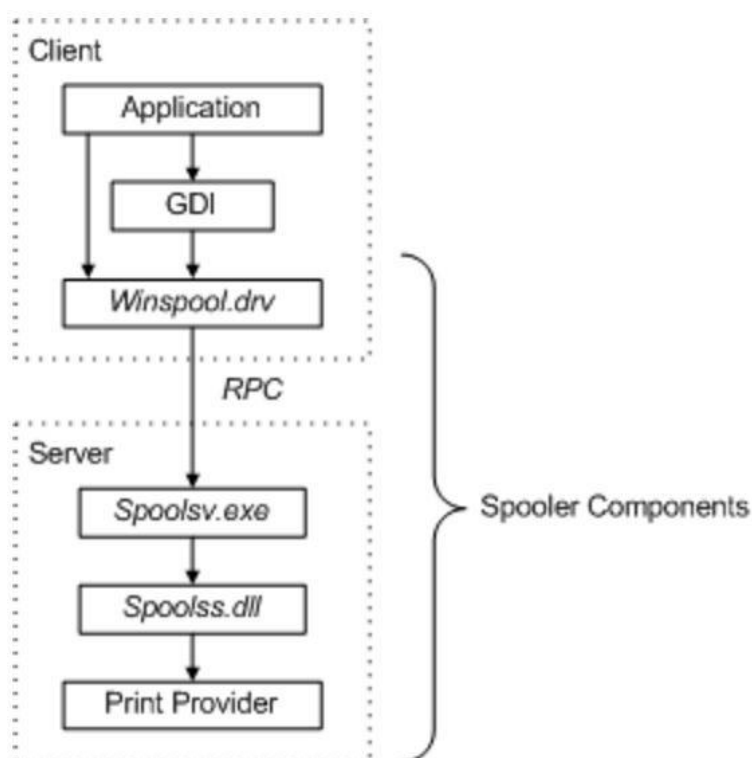


Ilustración 4 Sistema Impresión Windows

Consta de un cliente (client), que solicita los servicios de impresión a un servidor (server) para enviar un trabajo de impresión a una impresora.

El cliente consta de los siguientes elementos:

- **Aplicación:** para imprimir un documento se utiliza el componente de configuración proporcionada por el driver de la impresora con el que se configura las características y se crea el trabajo de impresión. Si para la creación del trabajo de impresión necesita soporte gráfico hace una llamada a las funciones del módulo GDI, en caso contrario envía el trabajo de impresión haciendo una llamada al Spooler a través de la interfaz Winspool.drv del cliente.
- **GDI (Graphics Device Interface):** Incluye dos componentes: user-mode y kernel-mode. El componente user-mode es el utilizado por la aplicación cuando requiere soporte gráfico, en cuyo caso, con la información recibida desde la aplicación el GDI interactúa con el driver de la impresora para generar la salida particular para esa impresora. El componente kernel-mode facilita servicios y funciones que pueden utilizar los drivers de dispositivos gráficos.
- **Winspool.drv:** Es la interfaz del cliente que hace posible la utilización de las API del Spooler y proporciona acceso al servidor a través de llamadas a procedimientos remoto RPC (Remote Procedure Call). Es utilizada principalmente por el GDI aunque las aplicaciones también utilizan alguna de sus funciones.

El servidor consta de los siguientes elementos:

- **Spoolsv.exe:** Es el servidor de las API del Spooler. Está implementado como un servicio de Windows. Se inicia en el arranque del sistema operativo y se mantiene en ejecución mientras el sistema operativo está en funcionamiento. Exporta una interfaz RPC del lado de la parte servidora de la API del Spooler. Los clientes incluyen Winspool.drv a nivel local y Win32spl.dll para conexión remota. Aunque el módulo implementa algunas de las funciones de la API, la mayoría de las llamadas se pasan a un proveedor de impresión mediante un enrutador.
- **Spoolss.dll(Enrutador):** Determina a que proveedor de impresión hay que llamar basado en el nombre de impresora o identificador de manejador pasando la llamada de función al proveedor de impresión correcto. Actúa como un router.
- **Print Provider(Proveedor de impresión):** Los proveedores de impresión son los encargados de dirigir los trabajos de impresión a dispositivos de impresión locales o remotos. Es también responsable de las operaciones de gestión de las colas de impresión. Implementan un conjunto común de capacidades accesibles a través de funciones de la API; estas funciones son llamadas por el enrutador (spoolss.dll).

La parte servidora (spoolsv.exe) se encuentra en el servidor donde está conectada localmente la impresora y debe poder ser accedida mediante llamada RPC desde la parte cliente (winspool.drv) en el sistema donde se ejecuta la aplicación.

Microsoft suministra los siguientes proveedores:

- **Localspl.dll:** Proveedor de impresión local. Se ocupa de los trabajos de impresión dirigidos a impresoras locales.
- **Win32spl.dll:** Proveedor de impresión de red. Se ocupa de los trabajos de impresión dirigidos a servidores remotos.
- **Nwprovau.dll:** Proveedor de impresión de Novell Netware.
- **Inetpp.dll:** Proveedor de impresión de trabajos de impresión enviados a una dirección URL.

Vamos a ver con más detalle los proveedores de impresión local y remoto.

2.1.1 Proveedor de impresión Local

En el siguiente diagrama podemos ver con más detalle los componentes del proveedor de impresión:

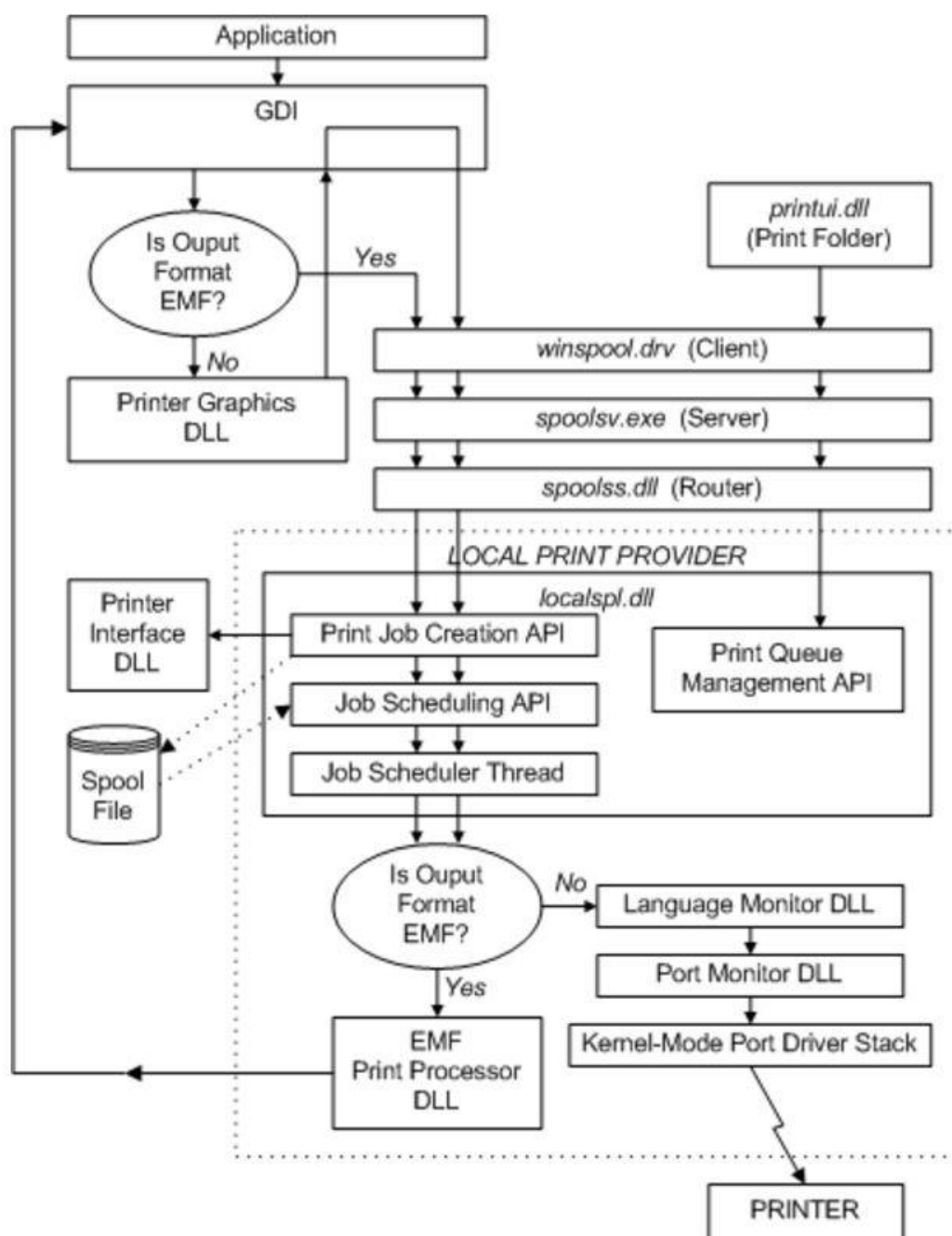


Ilustración 5 Sistema Impresión Windows. Detalle proveedor impresión local

El proveedor de impresión local facilita el control de los trabajos de impresión y la gestión de las impresoras locales, conectadas de forma local al sistema, accesibles a través del Monitor de Puertos (Port Monitor). A efectos prácticos las impresoras de red instaladas en el servidor también se configuran y acceden a través de un puerto local; las impresoras conectadas a sistemas operativos no pertenecientes a Microsoft, por ejemplo Novell, también son controladas a través de su correspondiente proveedor de forma local.

Como se muestra en el diagrama, una aplicación crea un trabajo de impresión llamando al Interfaz de dispositivos gráficos (GDI). Independientemente de si el formato inicial del trabajo de impresión es EMF(Enhanced MetaFile), La API de creación de trabajo de impresión, utilizando la DLL de la interfaz de impresora (drivers de impresora), crea un fichero de spool en disco y planifica su procesado. Una vez que se va a procesar el trabajo de impresión se lee el fichero de spool, si tiene formato EMF el procesador de impresión EMF envía de nuevo el trabajo de impresión al componente GDI para convertir el trabajo de impresión a formato RAW con la ayuda de una DLL para impresiones gráficas. El trabajo de impresión, ya en formato RAW, es enviado de vuelta a través del proveedor a la impresora sin tener que volver a planificarse de nuevo.

El proveedor de impresión local se encarga de enviar el flujo de datos de impresión al puerto de impresora adecuado.

Se utilizan dos tipos de Monitores de Impresión: Monitor de Lenguaje (Language monitor) y Monitor de Puerto (Port Monitor).

El Monitor de Lenguaje (DLL modo usuario) tiene dos funciones. En primer lugar proporciona información del estado de impresión bidireccionalmente entre el Spooler y la Impresora. En segundo lugar añade información de control al flujo de datos de impresión con comandos definidos en un lenguaje de control de trabajos de impresión, el más usual PJI, proporcionado por Microsoft a través del componente pjlmon.dll. El monitor de lenguaje es opcional y utilizado sólo por las impresoras que lo admiten.

Cuando una impresora admite la utilización de monitor de lenguaje, este recibe el flujo de datos de impresión desde el procesador de impresión, lo modifica, y se lo pasa al monitor del puerto de impresora.

El Monitor de Puerto de impresora consiste en una DLL (modo usuario) cuya función es proporcionar una vía de comunicación entre el Spooler y los drivers que acceden al puerto del hardware de E/S donde está conectada la impresora. Utiliza las funciones CreateFile, WriteFile, ReadFile y DeviceControl para comunicarse con los drivers del puerto. El Monitor de Puerto también proporciona las funciones necesarias para la gestión y configuración de los puertos de impresora de un servidor. Un puerto es la conexión física entre la cola de impresión y un único dispositivo de impresión.

2.1.2 Proveedor de Impresión Remoto

Un proveedor de impresión remoto se utiliza cuando la impresora a la que se lanza un trabajo de impresión no está conectada al servidor en el que se ejecuta la aplicación desde la que se quiere imprimir. En el siguiente diagrama se puede observar la distribución de componentes del sistema de impresión:

El esquema de funcionamiento es el siguiente:

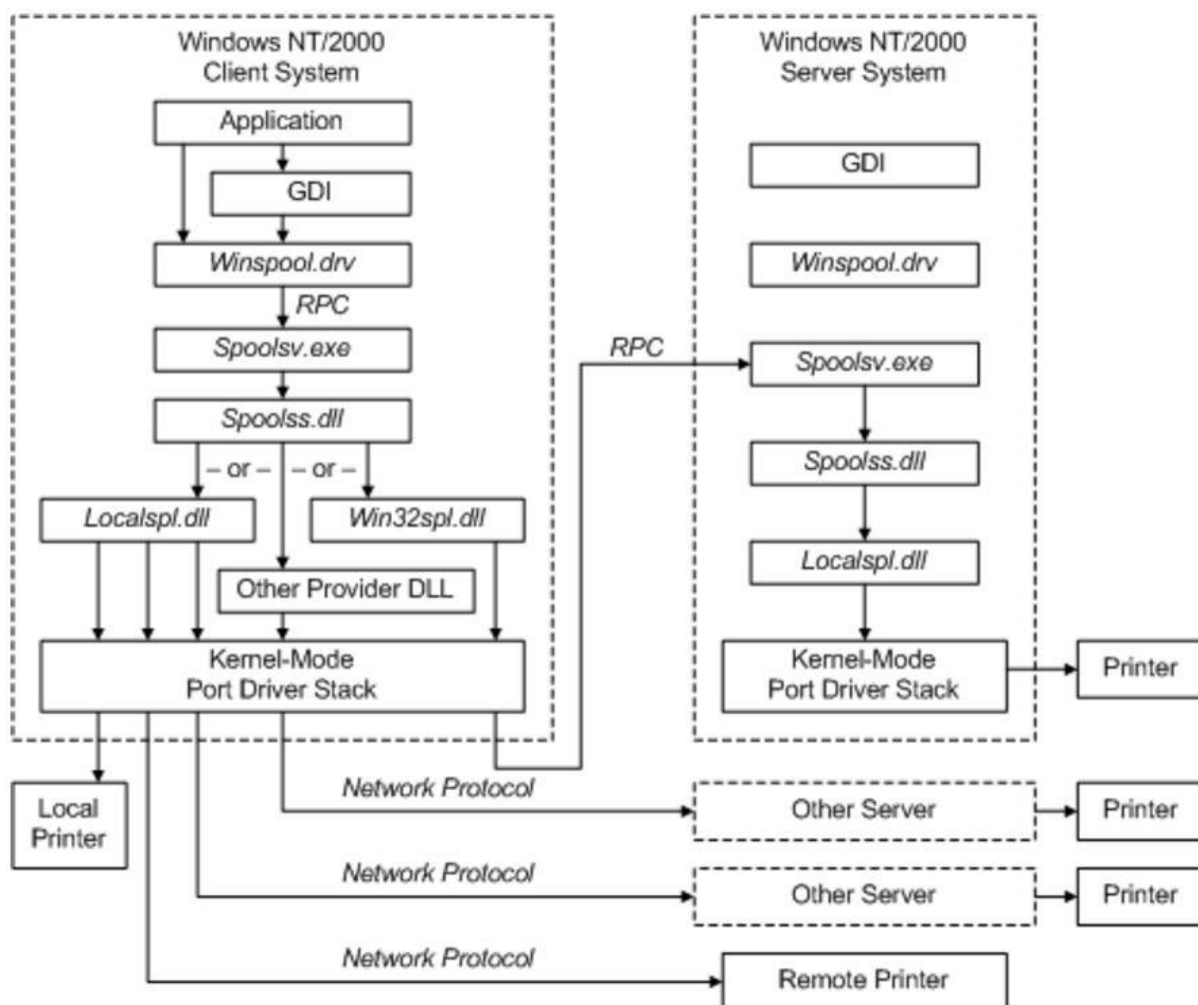


Ilustración 6 Sistema Impresión Windows. Detalle proveedor impresión remoto

El módulo GDI, o la aplicación directamente si no se requiere soporte gráfico, envía un trabajo de impresión utilizando la interfaz Winpool.drv del Spooler del lado del cliente. El servidor de impresión (Spoolsv.exe) consulta los proveedores disponibles y determina que se debe de utilizar un proveedor de impresión remoto a través del módulo Win32spl.dll, para ello realiza una llamada RPC a través del puerto adecuado del Servidor Local al Servidor de Impresión remoto. El Servidor del proveedor de impresión remoto recibe el flujo de datos de impresión y lo procesa de modo local como se ha explicado en el punto anterior.

La función básica del Spooler local es dirigir o enrutar el flujo de datos del cliente hacia el Spooler remoto donde está conectada la impresora por la que se quiere lanzar el trabajo de impresión.

2.1.3 Funciones definidas por los Proveedores de Impresión

El proveedor de impresión proporciona un conjunto completo de funciones. A continuación se presentan una muestra agrupadas por categorías:

- Función de inicialización de proveedor de impresión (InitializePrintProvider)
- Funciones de gestión de colas de impresión: Abrir impresora, cerrar impresora, añadir impresora, borrar Impresora, establecer propiedades impresora, obtener propiedades

impresora, enumerar impresoras, etc.(OpenPrinter, ClosePrinter, AddPrinter, DeletePrinter, SetPrinter, GetPrinter, EnumPrinters, etc.).

- Funciones de gestión de drivers de impresora: Añadir fichero con driver de impresora, borrar ficheros de drivers, listar los drivers de impresora del servidor, obtener información del driver de una impresora, etc.(AddPrinterDriver, DeletePrintDriver, EnumPrinterDrivers, GetPrinterDriver, etc.).
- Funciones de creación de Trabajos de impresión: Abortar , añadir, comenzar o terminar un trabajo de impresión, etc.(AbortPrinter, AddJob, StartDocPrinter, EndDocPrinter, etc.).
- Funciones de planificación de trabajos de impresión: Enumerar, consultar propiedades, modificar propiedades o estado de un trabajo de impresión.(EnumJobs, GetJob, SetJob, etc.).
- Funciones de gestión de formularios.
- Funciones de gestión del Procesador de Impresión: Instalar, borrar o enumerar procesadores de impresión, etc.(AddPrintProcessor, DeletePrintProcessor, EnumPrintProcessor, etc.).
- Funciones de gestión de Monitor de Impresión: Añadir, borrar o enumerar monitores de impresión, etc.(AddMonitor, DeleteMonitor, EnumMonitor, etc.).
- Funciones de gestión de puertos de impresora: Añadir, configurar, borrar, enumerar, etc.(AddPort, ConfigurePort, DeletePort, EnumPorts, etc.).
- Funciones de gestión del registro: Añadir, borrar y enumerar las claves de registro y los valores asociados con un proveedor de impresión. (SetPrinterData, DeletePrinterKey, EnumPrinterData).

2.1.4 Ampliación de la arquitectura de impresión a partir de Windows Vista: doble ruta.

A partir de Windows 2000 y en versiones anteriores a Windows vista, el sistema de impresión utilizaba la ruta de impresión GDI; este modelo se ha visto en los apartados anteriores. A partir de Windows Vista el sistema de impresión añade una nueva ruta de impresión basada en XPS (XML Paper Specification). En la siguiente ilustración se puede ver el modelo de doble ruta:

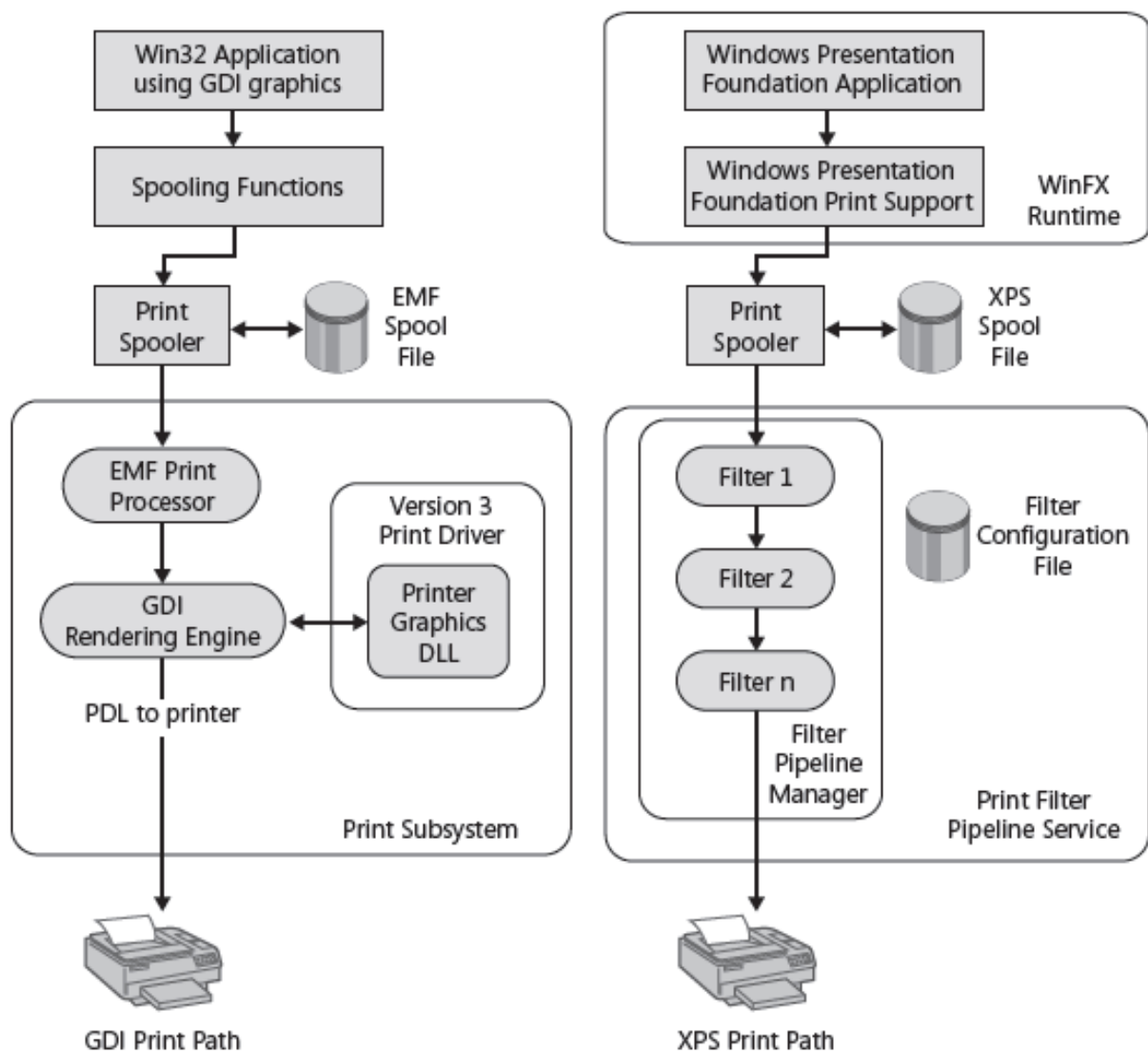


Ilustración 7 Arquitectura Sistema Impresión: doble ruta

Las novedades de la ruta de impresión XPS. Está basada en el modelo de controlador de impresora XPS (XPSDrv) y presenta las siguientes ventajas:

- Mantiene el formato de documento de impresión XPS desde que es generado por la aplicación hasta que es tratado por el controlador del dispositivo de impresión.
- La ruta de impresión XPS puede ser más eficiente proporcionando perfiles de color avanzados.
- Proporciona impresión WYSIWYG (what you see is what you get) lo que ves es lo que obtienes.
- Reducción del tamaño de la cola mediante diversos mecanismos:
 - Agrupación de fuentes en subconjuntos. En el archivo XPS sólo se almacenan los caracteres utilizados en el documento.
 - Compatibilidad con gráficos avanzados.
 - Identificación de recursos comunes: recursos utilizados varias veces se carga una sola vez (ejemplo una imagen que se repite).
 - Compresión ZIP. Todos los documentos XPS se comprimen en formato ZIP.

Cuando una aplicación WPF (Windows Presentation Foundation) lanza un trabajo de impresión hace una llamada al módulo de soporte de impresión WPF que tiene 2 funciones. En primer lugar crear un trabajo de impresión XPS y en segundo lugar hacer de interfaz del lado de cliente para hacer las llamadas a funciones del servidor de impresión spoolsv.exe.

Los drivers de impresión XPSDrv tienen las mismas funciones que los drivers de impresión GDI. Deben proporcionar una interfaz de usuario y se encargan de procesar el flujo de datos del trabajo de impresión y enviarlo a la impresora.

Con el nuevo modelo con doble ruta las aplicaciones pueden imprimir utilizando alguno de los dos modelos: GDI o XPS. La ruta de impresión se determina en función del tipo de controlador de impresora instalado en la cola de impresión de destino.

Si una aplicación Win32 envía un trabajo de impresión a una cola de impresión que utiliza un controlador de impresión basado en GDI se procesa utilizando la ruta de impresión GDI. Si una aplicación WPF genera un trabajo de impresión basado en una cola de impresión que utiliza un controlador de impresión XPSDrv, el trabajo se almacena en formato XPS y se procesa utilizando la ruta XPS. Pero también es posible que una aplicación Win32 lance un trabajo de impresión utilizando la ruta de impresión XPS y que una aplicación WPF lance un trabajo de impresión a través de la ruta GDI. Esto se consigue gracias a los módulos conversores entre rutas. La idea se puede ver en la siguiente ilustración:

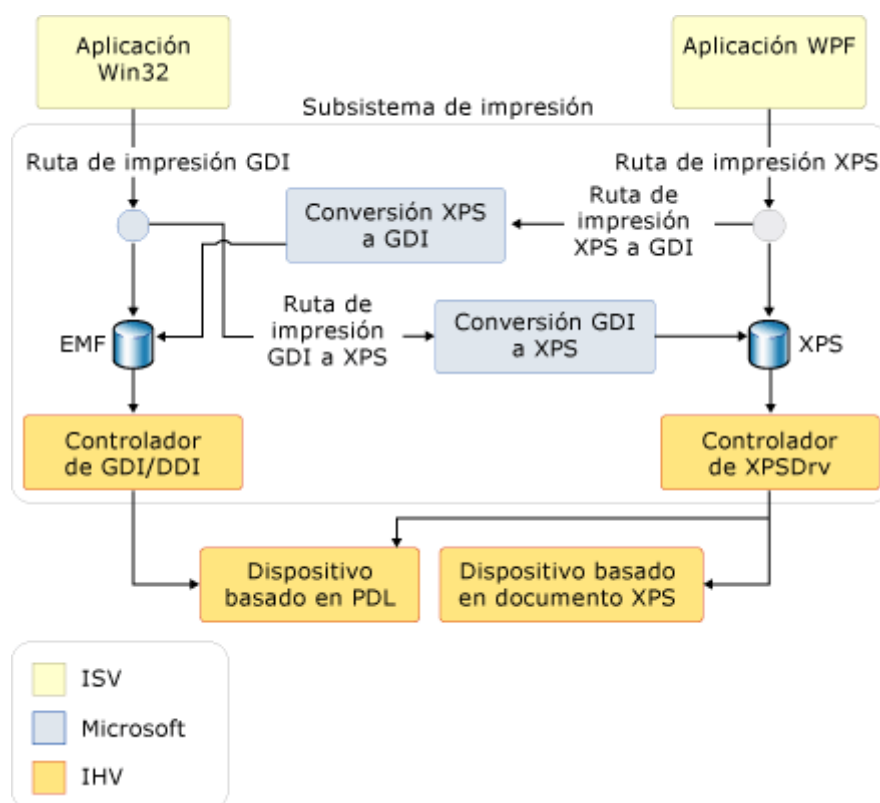


Ilustración 8 Arquitectura Sistema Impresión: doble ruta. Módulos conversores

WPF define una API básica, con las opciones de impresión más comunes, disponible a través del control de dialogo System.Windows.Controls.PrintDialog.

WPF también define una API avanzada con un conjunto completo de características XPS disponibles a través de los espacios de nombres System.Windows.Xps y System.Printing. Para Aplicaciones Win32 se recomienda la utilización del Espacio de nombres System.Drawing.Printing.

El espacio de nombres System.Printing.Interop proporciona conversión entre objetos PrintTicket administrados (ruta XPS) y estructuras de interfaz de dispositivo gráfico no administrador (ruta GDI) DEVMODE.

👁 *En ambas rutas es posible utilizar las API del interfaz winspool.drv para acceder a las funciones del Servidor de impresión.*

2.2 Obtención de información de los trabajos de impresión.

Vista la arquitectura del sistema de impresión Windows podemos identificar los posibles métodos para obtener los detalles de los trabajos de impresión. Estos son:

- Eventos del Servidor de Impresión.
- APIs del Servidor de Impresión.
- Namespace System.Printing
- Análisis de ficheros de spool del Sistema de Impresión Windows.

2.2.1 Eventos del Servidor de Impresión

Gracias al registro de eventos de Windows vamos a poder detectar cuando se está produciendo alguna actividad en el Sistema de Impresión por eso va a ser el punto de partida para poder obtener los detalles de los trabajos de impresión.

El Servicio de registro de eventos de Windows permite que los componentes del S.O, entre ellos el Servidor de impresión, y los programas instalados guarden registro de su actividad.

A partir de Windows Vista el Servicio de registro de eventos almacena la información en formato XML.

Los eventos de impresión, generados por el Servidor de Impresión, abren la puerta a todo el proceso de monitorización, ya que permiten detectar cuando se realiza una impresión para poder extraer sus propiedades. Además a través de los eventos de impresión ya se pueden obtener algunas propiedades importantes de los trabajos de impresión

Los eventos de impresión más importantes cuando se imprime un documento se muestran en la siguiente tabla:

Cód. Evento	Descripción	Propiedades
303	Pausar cola de impresión	✓ Impresora
304	Reanudar cola de impresión	✓ Impresora
306	Estableciendo configuración de impresora	✓ Impresora
307	Documento impreso	✓ Número de trabajo de impresión ✓ Nombre de documento ✓ Usuario ✓ Servidor de impresión ✓ Impresora ✓ Puerto ✓ Tamaño en bytes ✓ Número de páginas
309	Reanudando impresión	✓ Número de trabajo de impresión ✓ Nombre de documento ✓ Usuario

		✓ Impresora
310	Trabajo de impresión eliminado	✓ Número de trabajo de impresión ✓ Nombre de documento ✓ Usuario ✓ Nombre de impresora
800	Poniendo trabajo en cola	✓ Número de trabajo de impresión
801	Imprimiendo	✓ Número de trabajo de impresión
802	Eliminando trabajo de impresión	✓ Número de trabajo de impresión ✓ Tamaño en bytes ✓ Tipo datos ✓ Páginas ✓ Páginas por cara
805	Presentando trabajo de impresión	✓ Número de trabajo de impresión ✓ Tamaño en bytes ✓ Color ✓ Resolución horizontal ✓ Resolución vertical ✓ Calidad ✓ Copias
812	No se pueden eliminar ficheros de Spool	✓ Fichero de Spool ✓ Código de error
842	El servidor de impresión envió el trabajo a la impresora.	✓ Número de trabajo de impresión ✓ Procesador de impresión ✓ Impresora ✓ Driver

Tabla 1. Eventos de impresión

Es posible, desde una aplicación, suscribirse a los eventos del sistema de impresión y así obtener información de los trabajos de impresión, según la tabla de eventos anterior.

En concreto nos interesa el evento de impresión 307 que informa que un documento ha sido impreso.

A partir de esta información podemos identificar el trabajo de impresión (Número de trabajo de impresión, Impresora) y obtener más información utilizando las funciones proporcionadas por el Servidor de Impresión como se muestra en el siguiente punto.

2.2.2 APIs proporcionadas por el Servidor de impresión

Para más detalles se puede consultar “Print Spooler API functions” en el MSDN de Microsoft(“Print Spooler API Functions (Windows),” n.d.).

Podemos obtener las propiedades de un trabajo de impresión directamente a través de las APIs proporcionadas por el interfaz de cliente del Servidor de impresión: Winspool.drv.

De entre todas las funciones de las APIs del Servidor de impresión, las siguientes son suficientes para obtener los detalles de los trabajos de impresión y si es necesario obtener y modificar alguna propiedad de los dispositivos de impresión:

OpenPrinter (string *pPrinterName*, out IntPtr *phPrinter*, ref PRINTER_DEFAULTS *pPrinterDefaults*): API para interactuar con una impresora.

Parámetros:

<i>pPrinterName</i>	Nombre de la impresora
<i>phPrinter</i>	Identificador interno de impresora
<i>pPrinterDefaults</i>	Puntero a estructura PRINTER_DEFAULTS(características de la impresora)

ClosePrinter (IntPtr *hPrinter*): API para Cerrar la comunicación con una la impresora.

Parámetros:

<i>hPrinter</i>	Identificador interno de impresora (obtenido a través de OpenPrinter)
-----------------	---

GetPrinter (IntPtr *hPrinter*, uint *Level*, IntPtr *pPrinter*, uint *cbBuf*, out uint *pcbNeeded*): API para obtener las propiedades de una impresora.

Parámetros:

<i>hPrinter</i>	Identificador interno de impresora (obtenido a través de OpenPrinter)
<i>Level</i>	Nivel o tipo de estructura utilizada para obtener información de impresora. Entre 1-9 niveles.
<i>pPrinter</i>	Estructura de datos utilizada según el nivel que se indica en el parámetro Level. (Utilizamos la estructura:PRINTER_INFO_5
<i>cbBuf</i>	tamaño en bytes de la estructura apuntada por pPrinter Puntero a una variable donde la función guarda el tamaño, en bytes, de los datos con información de la impresora.

SetPrinter (IntPtr *hPrinter*, uint *Level*, IntPtr *pPrinter*, uint *Command*): API para modificar las propiedades de una impresora.

Parámetros:

<i>hPrinter</i>	Identificador interno de impresora (obtenido a través de OpenPrinter)
<i>Level</i>	Nivel o tipo de estructura utilizada para obtener información de

	impresora. Entre 1-9 niveles.
<i>pPrinter</i>	Estructura de datos utilizada según el nivel que se indica en el parámetro Level. (Utilizamos la estructura: PRINTER_INFO_5)
<i>Command</i>	Si el valor es 0. La impresora se reconfigura con los valores contenidos en la estructura apuntada por pPrinter

GetJob (IntPtr *hPrinter*, uint *JobId*, uint *Level*, IntPtr *pJob*, uint *cbBuf*, ref uint *pcbNeeded*):

API para obtener las propiedades de un trabajo de impresión.

Parámetros:

<i>hPrinter</i>	Identificador interno de impresora (obtenido a través de OpenPrinter)
<i>JobId</i>	Número de trabajo de impresión
<i>Level</i>	1:JOB_INFO_1, 2:JOB_INFO_2. Según el detalle que queramos obtener del trabajo de impresión. En concreto utilizo JOB_INFO_2 por que necesito obtener el máximo detalle.
<i>pJob</i>	Puntero a estructura indicada en Level.
<i>cbBuf</i>	tamaño en bytes de la estructura apuntada por pJob.
<i>pcbNeeded</i>	Puntero a una variable donde la función guarda el tamaño, en bytes, de los datos con información del trabajo de impresión.

2.2.3 Estructuras utilizadas en las funciones descritas:

JOB_INFO_2

```
typedef struct _JOB_INFO_2 {
    DWORD          JobId;
    LPTSTR         pPrinterName;
    LPTSTR         pMachineName;
    LPTSTR         pUserName;
    LPTSTR         pDocument;
    LPTSTR         pNotifyName;
    LPTSTR         pDataType;
    LPTSTR         pPrintProcessor;
    LPTSTR         pParameters;
    LPTSTR         pDriverName;
    LPDEVMODE      pDevMode;
    LPTSTR         pStatus;
    PSECURITY_DESCRIPTOR pSecurityDescriptor;
    DWORD          Status;
    DWORD          Priority;
    DWORD          Position;
    DWORD          StartTime;
    DWORD          UntilTime;
    DWORD          TotalPages;
    DWORD          Size;
    SYSTEMTIME      Submitted;
    DWORD          Time;
}
```

```

    DWORD          PagesPrinted;
}

```

Detalle 1. Estructura JOB_INFO_2

pDevMode apunta a una estructura _DEVMODE

```

typedef struct _DEVMODE {
    TCHAR  dmDeviceName[CCHDEVICENAME];
    WORD   dmSpecVersion;
    WORD   dmDriverVersion;
    WORD   dmSize;
    WORD   dmDriverExtra;
    DWORD  dmFields;
    union {
        struct {
            short dmOrientation;
            short dmPaperSize;
            short dmPaperLength;
            short dmPaperWidth;
            short dmScale;
            short dmCopies;
            short dmDefaultSource;
            short dmPrintQuality;
        };
        struct {
            POINTL dmPosition;
            DWORD  dmDisplayOrientation;
            DWORD  dmDisplayFixedOutput;
        };
    };
    short dmColor;
    short dmDuplex;
    short dmYResolution;
    short dmTTOption;
    short dmCollate;
    TCHAR  dmFormName[CCHFORMNAME];
    WORD   dmLogPixels;
    DWORD  dmBitsPerPel;
    DWORD  dmPelsWidth;
    DWORD  dmPelsHeight;
    union {
        DWORD  dmDisplayFlags;
        DWORD  dmNup;
    };
    DWORD  dmDisplayFrequency;
#ifdef WINVER >= 0x0400
    DWORD  dmICMMethod;
    DWORD  dmICMIntent;
    DWORD  dmMediaType;
    DWORD  dmDitherType;
    DWORD  dmReserved1;
    DWORD  dmReserved2;
#endif
#ifdef WINVER >= 0x0500 || (_WIN32_WINNT >= 0x0400)
    DWORD  dmPanningWidth;
    DWORD  dmPanningHeight;
#endif
#ifdef _WIN32_WINNT >= 0x0400
    DWORD  dmPanningWidth;
    DWORD  dmPanningHeight;
#endif
}

```

Detalle 2. Estructura DEVMODE

PRINTER_DEFAULTS

```
typedef struct _PRINTER_DEFAULTS {  
    LPTSTR    pDatatype;  
    LPDEVMODE pDevMode;  
    ACCESS_MASK DesiredAccess;  
}
```

Detalle 3. Estructura PRINTER_DEFAULTS

PRINTER_INFO_5

```
typedef struct _PRINTER_INFO_5 {  
    LPTSTR pPrinterName;  
    LPTSTR pPortName;  
    DWORD Attributes;  
    DWORD DeviceNotSelectedTimeout;  
    DWORD TransmissionRetryTimeout;  
}
```

Detalle 4. Estructura PRINTER_INFO5

2.2.4 Namespace System.Printing de .NET

También podemos interactuar con el Servidor de Impresión mediante el namespace System.Printing de .NET. Para obtener más información se puede consultar la documentación en el MSDN de Microsoft. ("System.Printing (Espacio de nombres)," n.d.)

El namespace System.Printing engloba un conjunto de clases con las que se puede gestionar Servidores, Colas y Trabajos de impresión.

Las clases principales son las siguientes:

- **LocalPrintServer:** Representa el Servidor de Impresión Local (el equipo donde se ejecuta la aplicación), permite administrar las colas de impresión. El método `GetPrintQueue` (pasándole como parámetro el nombre de impresora) devuelve un objeto `PrintQueue` asociado a la impresora especificada.
- **PrintQueue:** Representa las colas de impresión.
- **PrintSystemJobInfo:** Define en detalle un trabajo de impresión. Con el método de clase `Get` se puede obtener un objeto `PrintSystemJobInfo` (pasándole como parámetros el nombre de la impresora y el número de trabajo de impresión), con información de un trabajo de impresión lanzado por una impresora.

👁 *En el MSDN de Microsoft se hace la siguiente advertencia sobre el namespace System.Printing (MSDN Microsoft, n.d.-i). Texto Literal:*

"Classes within the System.Printing namespace are not supported for use within a Windows service or ASP.NET application or service. Attempting to use these classes from within one of these application types may produce unexpected problems, such as diminished service performance and run-time exceptions:"

Para el desarrollo del TFG será necesaria la creación de un servicio Windows que esté siempre activo capturando los trabajos de impresión. Pero a la vista de la advertencia en el MSDN hay que evitar siempre que sea posible el uso de este namespace.

2.2.5 Análisis de ficheros de Spool del Sistema de impresión de Windows.

El lenguaje de descripción de páginas (PDL: Page description Language) utilizado para guardar los trabajos de impresión en los ficheros de spool depende del driver del dispositivo de impresión.

Existen numerosos lenguajes PDL, pero sólo me voy a centrar en los utilizados en los dispositivos de impresión ofrecidos dentro del servicio de impresión del STIC; estos son: EMFSPool, EMF, JPL y PCLXL.

No me interesa profundizar en el análisis de estos lenguajes PDL hasta el nivel de ser capaz de reproducir los documentos representados en los ficheros de Spool. El análisis que voy a realizar consiste en poder identificar los elementos constitutivos del lenguaje PDL empleado y analizar aquellos elementos que me aportan información relevante para obtener algunas de las propiedades de los trabajos de impresión que no es posible obtener a través de las API del Servicio de Impresión.

👁 *La obtención de las propiedades de los trabajos de impresión a través del análisis de los ficheros de spool es dependiente del lenguaje PDL utilizado. Es el más costoso de implementar. Además requiere de la implementación de rutinas específicas para cada lenguaje PDL que se necesita analizar.*

2.2.5.1 PJP (Printer Job Language)

Realmente no se trata de un lenguaje PDL si no de un lenguaje desarrollado con el propósito de proporcionar un nivel de control sobre los trabajos de impresión que no es posible conseguir con otros lenguajes PDL. Puede simular las funciones de panel de control que anteriormente sólo podían ser controladas a través del panel de control físico del dispositivo de impresión.

Para conseguir este control se utilizan comandos PJP como una mejora a los lenguajes PDL, como puede ser PCLXL o HP-GL2. Proporciona cuatro funciones principales:

- Cambiar el lenguaje de impresión entre trabajos de impresión
- Separación de trabajos
- Configurar las impresoras
- Lectura de retorno del estado de la impresora desde la impresora hasta el ordenador anfitrión

Dentro del sistema de impresión los trabajos se almacenan en ficheros de Spool utilizando: PJP+ un lenguaje PDL.

Por ejemplo para la impresora LaserJet 5500 del Servicio de impresión del STIC, los trabajos de impresión se almacenan en ficheros de Spool PJP/PCLXL y para el plotter HP Designjet 800 los trabajos de impresión se almacenan en ficheros de spool PJP/HPGL2

En principio para el objetivo de este TFG es interesante identificar el contenido PJP de los ficheros de Spool, es decir donde empieza y donde acaba el bloque PJP y que comandos PJP contiene, analizando aquellos que son de interés. Por ejemplo el comando PJP que conmuta de lenguaje PJP a otro lenguaje PDL como PCLXL.

2.2.5.1.1 Sintaxis de las líneas de comandos PJP

En la siguiente tabla se muestran las convenciones y códigos de control utilizados para expresar la sintaxis de las instrucciones PJP.

COMANDOS	Las palabras en mayúsculas indican nombres de comandos PJP y palabras que se escriben textualmente.
[]	Indica que el contenido es opcional.
<>	Para identificar que el contenido que encierra es un carácter tipo código de control
<LF>	Carácter de avance de línea (ASCII 0x0a)
<SP>	Carácter espacio (ASCII 0x20)
<ESC>	Carácter Escape (ASCII 0x1b)
<CR>	Carácter retorno de carro (ASCII 0x1d)

Tabla 2. Sintaxis instrucción PJP

Hay 4 formatos de sintaxis PJP; excepto el primer formato, en el resto todas las instrucciones comienzan por “@PJP” y finaliza por el carácter “<LF>”.

Formato #1: Es la única línea de comando que tiene este formato y se utiliza para identificar el principio y el final del contenido JPL. Todos los trabajos PJJL deben empezar y terminar con el comando UEL (The Universal Exit Language command):

<ESC>%-12345X

Formato #2: Permite introducir una línea PJJL sin ningún comando. Se utiliza para añadir claridad a un listado de comandos JPL.

@PJJL [<CR>]<LF>

Formato #3: Sólo se utiliza para los comandos ECHO y COMMENT. Sirve para introducir comentarios.

@PJJL<SP>command<SP>[<words>][<CR>]<LF>

Formato #4: El resto de comandos JPL utiliza este formato.

@PJJL<SP>command<SP>[command modifier:value] [option name [= value]][<CR>]<LF>

Donde:


Command: representa un nombre de comando JPL, como SET, ENTER, EOJ, JOB, etc.

[Command modifier : value]: Es un modificador de comando que permite especificar qué está afectado por el comando. Un comando PJJL sólo puede contener un modificador de comando en cada instrucción PJJL. Por ejemplo el modificador de comando JOB: "NAME", sirve para establecer el nombre del trabajo de impresión:

@PJJL JOB NAME="Nombre de trabajo de impresión"<LF>

[option name [= value]]: El parámetro opción (Option) sirve para especificar una opción o conjunto de opciones de commando. Hay opciones de comando que requieren un valor. La siguiente instrucción estable el tamaño por defecto en 14.25:

@PJJL DEFAULT LPARM:PCL PTSIZE=14.25<LF>

 *Algunas reglas de sintaxis que hay que tener en cuenta:*

- El prefijo PJJL "@PJJL" siempre debe estar en mayúsculas
- Entre el prefijo @PJJL y el nombre del comando debe haber siempre un espacio en blanco, también entre el nombre de comando y los modificadores de comando, por ejemplo:

@PJJL ENTER LENGUAJE=PCLXL<LF>

- Tampoco se permite espacios en blanco entre carácter retorno de carro <CR> y el carácter retorno de línea <LF>
- Todos los trabajos JPL deben empezar y terminar con el comando UEL: "<ESC>%-12345X". Este comando indica que se conmuta del actual lenguaje de impresión a lenguaje PJJL.
- Al comando UEL del comienzo de un trabajo debe seguirle inmediatamente el prefijo de comando PJJL (@PJJL).
- Todas las instrucciones PJJL deben terminar con el carácter fin de línea (<LF>), excepto el comando UEL.

2.2.5.1.2 Tipos de variables

En PjL se utilizan variables alfanuméricas, numéricas y de cadena donde los caracteres (dígitos, letras y símbolos) están referidos a la tabla de caracteres Roman-8 (CP1051: Code Page 1051)

Alfanuméricas: Combinación de letras y dígitos, donde el primer carácter siempre tiene que ser una letra. Las letras pueden ser mayúsculas (caracteres ASCII del 0x41 al 0x5a), minúsculas (caracteres ASCII del 0x61 al 0x7a) y dígitos (caracteres ASCII del 0x30 al 0x39).

Ejemplo: LaserJet800

Numéricas: Cualquier número está compuesto de dígitos, y opcionalmente, un punto decimal (ASCII 0x2e), y los signos “+” y “-”(ASCII 0x2b y 0x2d respectivamente).

Ejemplo: LaserJet800

-111.111

0.123456

Cadenas: Las cadenas deben ir entrecomilladas. Consisten en cualquier combinación de caracteres ASCII desde 0x20 hasta el 0xff, añadiendo también el carácter ASCII de tabulación horizontal 0x09 y excluyendo el carácter ASCII de comillas dobles 0x22.

Ejemplo:

“Esto es una cadena valida con mas de 30 caracteres”

Se pueden agrupar los comandos PjL por su funcionalidad en los siguientes grupos:

Comandos Kernel (Kernel Commands): Entre los comandos de este grupo se encuentra el comando UEL y el comando ENTER con el que se selecciona el lenguaje de impresión para procesar el trabajo de impresión.

Comandos para separar trabajos (Job Separation Commands): Este grupo de comandos se utiliza para marcar los límites, indicando donde comienza y acaba un trabajo PjL.

Comandos de entorno (Environment Commands): Con este grupo de comandos se pueden establecer características del dispositivo de impresión. En concreto el comando SET establece muchas propiedades aplicadas a un trabajo de impresión que se pueden obtener analizando el contenido PjL del fichero de spool de impresión; por ejemplo el número de copias, nombre del documento, usuario que lanza el trabajo, desde que equipo, etc.

Comandos de consulta de estado (Status Readback Commands): Permite obtener información de la configuración y estado de la impresora. Por ejemplo si una bandeja está abierta, si el nivel de tóner es bajo, si está en estado online/offline, etc.

Comandos de asistencia del dispositivo (Device Attendance Commands): Permite cambiar los mensajes mostrados en el panel del dispositivo de impresión. Por ejemplo para alertar de que la impresora se ha quedado sin papel y que hay que avisar al operador.

Comandos PjL de sistema de archivos (PjL FileSystem Commands): Este grupo de comandos es útil para dispositivos de impresión que disponen de un dispositivo de almacenamiento masivo permitiendo interactuar con su sistema de archivos (FSAPPEND, FSDIRLIST, FSDELETE, FSMKDIR, etc.)

2.2.5.2 PCLXL

Basado en la documentación de Hewlett Packard *PCL XL Feature Reference*(Packard, 2002; “PCL XL Feature Reference Protocol Class 2.0 Revision: p2.2 Word for Windows File: xl_ref20r22.doc Revision Date: March16,2000 Word for Windows Version: Word97 Author(s): FIGURE FILES: VISIO 4.0,” n.d.).

HP creo el lenguaje de descripción de páginas PCL a principio de los años 80 para impresoras matriciales y de chorro de tinta. Se han ido sacando nuevas versiones del lenguaje original aportando nuevas mejoras. HP lanzo su primera impresora Láser (HP Laser Jet) en 1984 con versión PCL 3. La versión actual es PCL6 o PCLXL, su primer lanzamiento se produce a finales de 1995.

Hasta la versión 5, el lenguaje PCL estaba basado en secuencias de escape y las versiones superiores incluían las secuencias de escape de las versiones precedentes. La versión 6 de PCL o PCLXL es en realidad un nuevo lenguaje de impresión.

El lenguaje PCLXL consiste en un conjunto de operadores con el que es posible representar páginas con cualquier texto y contenido gráfico. Este conjunto de operadores está pensado para poder utilizar todas las capacidades de los dispositivos gráficos actuales y poder extender sus capacidades en el futuro.

PCLXL es ampliamente utilizado en dispositivos de impresión HP. Otros fabricantes como Canon, Ricoh, Samsung, Brother, por citar algunos, también ofrecen dispositivos de impresión que admiten lenguaje PCLXL. Para estos dispositivos el Sistema de Impresión Windows almacena los trabajos de impresión en los ficheros de Spool utilizando lenguaje PCLXL.

El lenguaje PCLXL se puede ver como una sucesión de operadores que pueden llevar asociados una serie de atributos y de datos embebidos en el propio operador. Está diseñado para tener el mínimo número de operadores para su facilidad de uso, compatibilidad, eficiencia y rendimiento. Un operador puede utilizar un número variable de atributos y datos embebidos. Los atributos utilizan distintos tipos de datos.

Se puede esquematizar de la siguiente manera:

LenguajePCLXL ::= { DatosEmbebidos_{m>=0} + ListaAtributos + Operador }_n

Donde ListaAtributos tiene la siguiente estructura:

ListaAtributos ::= { TipoDato + Valor + IdAtributo }_{n>=0}

Donde el subíndice “m>=0” significa que puede haber desde 0 a m ocurrencias del elemento.

El mismo significado para el subíndice “n>=0”.

2.2.5.2.1 Operadores

Los operadores se identifican mediante un valor binario de un byte. La siguiente enumeración contiene el conjunto de operadores PCLXL.

```
enum OPERADOR
{
    BeginSession = 0x41,
    EndSession = 0x42,
    BeginPage = 0x43,
    EndPage = 0x44,
```

VendorUnique = 0x46,
Comment = 0x47,
OpenDataSource = 0x48,
CloseDataSource = 0x49,
EchoComment = 0x4a,
Query = 0x4b,
BeginFontHeader = 0x4f,
ReadFontHeader = 0x50,
EndFontHeader = 0x51,
BeginChar = 0x52,
ReadChar = 0x53,
EndChar = 0x54,
RemoveFont = 0x55,
SetCharAttributes = 0x56,
SetDefaultGS = 0x57,
SetColorTreatment = 0x58,
BeginStream = 0x5b,
ReadStream = 0x5c,
EndStream = 0x5d,
ExecStream = 0x5e,
RemoveStream = 0x5f,
PopGS = 0x60,
PushGS = 0x61,
SetClipReplace = 0x62,
SetBrushSource = 0x63,
SetCharAngle = 0x64,
SetCharScale = 0x65,
SetCharShear = 0x66,
SetClipIntersect = 0x67,
SetClipRectangle = 0x68,
SetClipToPage = 0x69,
SetColorSpace = 0x6a,
SetCursor = 0x6b,
SetCursorRel = 0x6c,
SetHalftoneMethod = 0x6d,
SetFillMode = 0x6e,
SetFont = 0x6f,
SetLineDash = 0x70,
SetLineCap = 0x71,
SetLineJoin = 0x72,
SetMiterLimit = 0x73,
SetPageDefaultCTM = 0x74,
SetPageOrigin = 0x75,
SetPageRotation = 0x76,
SetPageScale = 0x77,
SetPatternTxMode = 0x78,
SetPenSource = 0x79,
SetPenWidth = 0x7a,
SetROP = 0x7b,
SetSourceTxMode = 0x7c,
SetCharBoldValue = 0x7d,
SetNeutralAxis = 0x7e,
SetClipMode = 0x7f,
SetPathToClip = 0x80,
SetCharSubMode = 0x81,
CloseSubPath = 0x84,
NewPath = 0x85,
PaintPath = 0x86,
ArcPath = 0x91,
SetColorTrapping = 0x92,
BezierPath = 0x93,
SetAdaptiveHalfto = 0x94,

```
BezierRelPath = 0x95,  
Chord = 0x96,  
ChordPath = 0x97,  
Ellipse = 0x98,  
EllipsePath = 0x99,  
LinePath = 0x9b,  
LineRelPath = 0x9d,  
Pie = 0x9e,  
PiePath = 0x9f,  
Rectangle = 0xa0,  
RectanglePath = 0xa1,  
RoundRectangle = 0xa2,  
RoundRectanglePat = 0xa3,  
Text = 0xa8,  
TextPath = 0xa9,  
SystemText = 0xaa,  
BeginImage = 0xb0,  
ReadImage = 0xb1,  
EndImage = 0xb2,  
BeginRastPattern = 0xb3,  
ReadRastPattern = 0xb4,  
EndRastPattern = 0xb5,  
BeginScan = 0xb6,  
EndScan = 0xb8,  
ScanLineRel = 0xb9,  
PassThrough = 0xbf  
}
```

Detalle 5. Enumeración con operadores PCLXL

A Continuación se agrupan los operadores por funcionalidades. No se va a entrar a describir los operadores en detalle con los atributos que admite cada uno de ellos. Para el caso de estudio que nos ocupa es suficiente poder identificar todos los operadores y centrarnos con más detalle en aquellos operadores que nos van a aportar información para el desarrollo del TFG, como por ejemplo, aquellos operadores necesarios para identificar las páginas de un documento y en qué páginas de un documento aparece contenido en color. De todos modos el identificador utilizado para cada operador suele ser lo bastante explicativo.

2.2.5.2.1.1 Operadores de Sesión

Este conjunto de operadores permiten definir los límites de la sesión en la que se define un documento, las páginas que componen el documento y los orígenes de datos necesarios para realizar algunas de las operaciones en las que no es posible proporcionar datos a un operador a través de un conjunto de atributos.

OperadoresSesión ::= {**BeginSession** & **EndSession** & **BeginPage** & **EndPage** & **Comment** & **OpenDataSource** & **CloseDataSource**}

Un documento con “n” páginas representado con PCLXL tiene la siguiente estructura:

{BeginSession + {BeginPage + EndPage }_n + EndSession}

2.2.5.2.1.2 Operadores de control de fuentes

Este conjunto de operadores permiten definir caracteres de un tipo de fuente (TrueType, Bitmap, etc.) en cualquier lugar del documento. Cada carácter se trata como un objeto gráfico independiente.

OperadoresFuentes ::= {**BeginFontHeader & ReadFontHeader & EndFontHeader & BeginChar & ReadChar & EndChar & RemoveFont & SetFont**}

2.2.5.2.1.3 Operadores de estado para gráficos

Permiten establecer y obtener los valores de los atributos de estado gráfico. Todos los atributos de estado gráfico se establecen con un valor por defecto cada vez que se define una nueva página con el operador BeginPage.

Como ejemplo de atributos de estado

Atributo de estado Gráfico	Descripción	Valor defecto
BrushSource	Color asociado a la brocha actual	RGBBlack
ColorSpace	Espacio de color utilizado.	RGB
PenSource	Color actual de la pluma.	RGBBlack
PenWidth	Anchura del trazo de la pluma	1 unidad usuario
....		

Tabla 3. PCLXL. Atributos de estado gráfico

Casi todos los operadores de estado están precedidos por “**Set...**”

La siguiente regla recoge todos los operadores de estado gráfico:

OperadoresEstadoGrafico ::= {**PopGS & PushGS & SetCursor & SetCursorRel & SetColorSpace & SetBrushSource & SetPenSource & SetCharAngle & SetCharScale & SetCharShear & SetCharBoldValue & SetCharSubMode & SetCharAttributes & SetFont & SetFillMode & SetMiterLimit & SetLineCap & SetLineJoin & SetLineDash & SetPainWidth & SetClipReplace & SetClipIntersect & SetClipRectangle & SetClipToPage & SetPathToClip & SetClipMode & SetPaintTxMode & SetSourceTxMode & SetRop & SetHalftoneMethod & SetPageDefaultCTM & SetPageOrigin & SetPageRotation & SetPageScale & SetHalftoneMethod & SetAdaptiveHalftoning & SetColotTrapping & SetNeuralAxis & SetDefaultGS & SetColorTreatment**}

Estos operadores se pueden agrupar en las siguientes categorías:

- Operadores para guardar y restaurar estados gráficos::= {**PopGS & PushGS**}
- Operadores para establecer y cambiar la localización del cursor ::= {**SetCursor & SetCursorRel**}
- Operadores para establecer el espacio de color y el color de los objetos de dibujo ::= { **SetColorSpace & SetBrushSource & SetPenSource** }
- Operadores para establecer fuentes y atributos de los caracteres ::= { **SetCharAngle & SetCharScale & SetCharShear & SetCharBoldValue & SetCharSubMode & SetCharAttributes & SetFont** }
- Operadores para establecer los atributos de la actual Ruta ::= { **SetFillMode & SetMiterLimit & SetLineCap & SetLineJoin & SetLineDash & SetPainWidth** }
- Operadores para establecer áreas de dibujo ::= { **SetClipReplace & SetClipIntersect & SetClipRectangle & SetClipToPage & SetPathToClip & SetClipMode**}
- Los operadores lógicos establecen cómo interactúan las imágenes que se tienen que representar con las imágenes ya existentes en el destino ::= {**SetPaintTxMode &**

SetSourceTxMode & SetRop & SetHalftoneMethod & SetPageDefaultCTM & SetPageOrigin & SetPageRotation & SetPageScale}

2.2.5.2.1.4 Operadores de dibujo

Los operadores de dibujo permiten dibujar caminos, Arcos, líneas, líneas Bezier, elipses, rectángulos, imágenes de mapas de bit, texto, etc.

Operadores de dibujo ::= { **CloseSubPath & NewPath & PatinPath & ArcPath & BezierPath & BezierPath & Chord & ChordPath & Ellipse & EllipsePath & LinePath & LineRelPath & Pie & PiePath & Rectangle & RectanglePath & RoundRectangle & RoundRectanglePath & BeginImage & ReadImage & EndImage & BeginRastPattern & ReadRastPattern & EndRastPattern & BeginScan & ScanLineRel & EndScan & Text & TextPath** }

2.2.5.2.1.5 Operadores de flujo de datos PCLXL (Streams).

Un dispositivo PCLXL puede recibir una secuencia de operadores y datos PCLXL identificados bajo un nombre, es lo que se denomina como flujo de datos definido por el usuario (User-Defined Stream). Esta secuencia de “operadores + datos” se carga en el dispositivo PCLXL mediante los operadores BeginStream/ReadStream/EndStream y puede ser referenciada por su nombre más tarde para su ejecución mediante el operador ExecStream.

Un flujo de datos PCLXL puede ser ejecutado en cualquier lugar, dentro de una sesión definida por los operadores BeginSession/EndSession, con el operador ExecStream.

Operadores de flujo de datos PCLXL ::= { **BeginStream & ReadStream & EndStream & ExecStream & RemoveStream & PassThrough** }

Los flujos de datos PCLXL definidos por el usuario tienen la siguiente estructura:

<p>Cabecera de flujo de datos (Stream Header) (Contiene información del Stream)</p>
<p>Cuerpo del flujo de datos (Stream Body) (Operadores y datos)</p>

La cabecera de un flujo de datos PCLXL contiene información de dicho flujo de datos. La cabecera tiene la siguiente estructura:

Byte inicial	Byte final	Descripción del elemento de la cabecera
0	0	Identificador de formato de representación (Binding format Identifier).
1	1	Reservado, carácter espacio en ASCII (0x20)
2	N-1	Cadena de descripción del flujo de datos (Stream Descriptor String).
N	N	Fin de cabecera. Utiliza el carácter ASCII para avance de línea (lf: 0x0a)
N+1	N+1	Comienza el cuerpo del flujo de datos PCLXL

Tabla 4. Cabecera PCLXL

Identificador de formato de representación (**Binding format Identifier**): Es un byte con el que se indica que reglas se siguen para representar los valores de los operadores y datos del cuerpo del flujo de datos, según se indica en la siguiente tabla:

Carácter	Valor Hexadecimal	Significado Identificador
'	0x27	Carácter comilla simple. Se utilizan caracteres ASCII para identificar operadores y datos.
)	0x29	Carácter paréntesis derecho. Los operadores, atributos y valores se expresan utilizando codificación Little Endian.
(0x28	Carácter paréntesis izquierdo. Los operadores, atributos y valores se expresan utilizando codificación Big Endian.

Tabla 5. Cabecera PCLXL. Identificador de formato.

Cadena de descripción del flujo de datos (**Stream Descriptor String**): Es una cadena de texto que acaba con un carácter salto de línea (line-feed: **If**. Valor hexadecimal 0x0a). La cadena debe tener al menos los 3 siguientes campos: (1) el primer campo indica el nombre de la clase de stream utilizado, (2) el segundo campo indica la versión de los operadores de la clase utilizada en el cuerpo del flujo de datos y (3) el tercer campo indica la revisión dentro de la versión utilizada. Los campos están separados por “punto y coma” y los valores por “coma”.

La siguiente ilustración muestra el detalle de la siguiente cabecera:

) HP-PCL XL;2;1;Comment Copyright(c) 1999 Microsoft Corporation<If>

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
00000000	29	20	48	50	2d	50	43	4c	20	58	4c	3b	32	3b	31	3b) HP-PCL XL;2;1;
00000010	43	6f	6d	6d	65	6e	74	20	43	6f	70	79	72	69	67	68	Comment Copyrigh
00000020	74	28	63	29	20	31	39	39	39	20	4d	69	63	72	6f	73	t(c) 1999 Micros
00000030	6f	66	74	20	43	6f	72	70	6f	72	61	74	69	6f	6e	0a	oft Corporation.

Ilustración 9 Ejemplo de Stream Header.

Se puede observar como la cadena finaliza con el carácter ASCII para avance de línea **0x0a** (posición 0x3f).

👁 En los ficheros de Spool de impresión que utilizan lenguaje JPL/PCLXL (JPL + PCLXL) para guardar los trabajos de impresión también se utiliza una cabecera, con la misma estructura, antes del comienzo de la sesión en la que se define el documento a imprimir. Se puede esquematizar de la siguiente manera:

@PJL ENTER LANGUAGE=PCLXL	//PJL: Indica el comienzo de contenido PCLXL.
Stream Header PCLXL	//Cabecera PCLXL.
Session PCLXL	//Sesión PCLXL con el contenido del documento.
ESC%-12345X	//PJL: Indica que termina el contenido PCLXL.

El cuerpo de un flujo de datos definido por el usuario contiene un conjunto de operadores y datos.

flujoPCLXL ::= { DatosEmbebidos_{n ≥ 0} + ListaAtributos + Operador }_n

2.2.5.2.2 Atributos y tipos de datos

Las siguientes enumeraciones contienen el conjunto de atributos y tipos de datos disponibles en PCLXL.

```
enum ATRIBUTO
{
    PaletteDepth = 0x02,
    ColorSpace = 0x03,
    NullBrush = 0x04,
    NullPen = 0x05,
    PaletteData = 0x06,
    PaletteIndex = 0x07,
    PatternSelectID = 0x08,
    GrayLevel = 0x09,
    RGBColor = 0x0b,
    PatternOrigin = 0x0c,
    NewDestinationSize = 0x0d,
    PrimaryArray = 0x0e,
    PrimaryDepth = 0x0f,
    AllObjectTypes = 0x1d,
    TextObjects = 0x1e,
    VectorObjects = 0x1f,
    RasterObjects = 0x20,
    DeviceMatrix = 0x21,
    DitherMatrixDataType = 0x22,
    DitherOrigin = 0x23,
    MediaDestination = 0x24,
    MediaSize = 0x25,
    MediaSource = 0x26,
    MediaType = 0x27,
    Orientation = 0x28,
    PageAngle = 0x29,
    PageOrigin = 0x2a,
    PageScale = 0x2b,
    ROP3 = 0x2c,
    TxMode = 0x2d,
    CustomMediaSize = 0x2f,
    CustomMediaSizeUnits = 0x30,
    PageCopies = 0x31,
    DitherMatrixSize = 0x32,
    DitherMatrixDepth = 0x33,
    SimplexPageMode = 0x34,
    DuplexPageMode = 0x35,
    DuplexPageSide = 0x36,
    ArcDirection = 0x41,
    BoundingBox = 0x42,
    DashOffset = 0x43,
    EllipseDimension = 0x44,
    EndPoint = 0x45,
    FillMode = 0x46,
    LineCapStyle = 0x47,
    LineJoinStyle = 0x48,
    MiterLength = 0x49,
    LineDashStyle = 0x4a,
    PenWidth = 0x4b,
    Point = 0x4c,
    NumberOfPoints = 0x4d,
    SolidLine = 0x4e,
    StartPoint = 0x4f,
    PointType = 0x50,
    ControlPoint1 = 0x51,
```

ControlPoint2 = 0x52,
ClipRegion = 0x53,
ClipMode = 0x54,
ColorDepthArray = 0x61,
ColorDepth = 0x62,
BlockHeight = 0x63,
ColorMapping = 0x64,
CompressMode = 0x65,
DestinationBox = 0x66,
DestinationSize = 0x67,
PatternPersistence = 0x68,
PatternDefineID = 0x69,
SourceHeight = 0x6b,
SourceWidth = 0x6c,
StartLine = 0x6d,
PadBytesMultiple = 0x6e,
BlockByteLength = 0x6f,
NumberOfScanLines = 0x73,
PrintableArea = 0x74,
TumbleMode = 0x75,
ContentOrientation = 0x76,
FeedOrientation = 0x77,
ColorTreatment = 0x78,
CommentData = 0x81,
DataOrg = 0x82,
Measure = 0x86,
SourceType = 0x88,
UnitsPerMeasure = 0x89,
QueryKey = 0x8a,
StreamName = 0x8b,
StreamDataLength = 0x8c,
PCLSelectFont = 0x8d,
ErrorReport = 0x8f,
VUExtension = 0x91,
VUDataLength = 0x92,
VUAttr1 = 0x93,
VUAttr2 = 0x94,
VUAttr3 = 0x95,
VUAttr4 = 0x96,
VUAttr5 = 0x97,
VUAttr6 = 0x98,
VUAttr7 = 0x99,
VUAttr8 = 0x9a,
VUAttr9 = 0x9b,
VUAttr10 = 0x9c,
VUAttr11 = 0x9d,
VUAttr12 = 0x9e,
EnableDiagnostics = 0xa0,
CharAngle = 0xa1,
CharCode = 0xa2,
CharDataSize = 0xa3,
CharScale = 0xa4,
CharShear = 0xa5,
CharSize = 0xa6,
FontHeaderLength = 0xa7,
FontName = 0xa8,
FontFormat = 0xa9,
SymbolSet = 0xaa,
TextData = 0xab,
CharSubModeArray = 0xac,
WritingMode = 0xad,
BitmapCharScaling = 0xae,

```

XSpacingData = 0xaf,
YSpacingData = 0xb0,
CharBoldValue = 0xb1,
}

```

Detalle 6. Enumeración con atributos PCLXL

```

enum TIPODATO
{
    ubyte = 0xc0, //1 byte
    uint16 = 0xc1, // 2 bytes
    uint32 = 0xc2, // 4 bytes
    sint16 = 0xc3, // 2 bytes
    sint32 = 0xc4, // 4 bytes
    real32 = 0xc5, // 4 bytes
    ubyte_array = 0xc8, // Array de elementos de 1 byte
    uint16_array = 0xc9, // Array de elementos de 2 bytes
    uint32_array = 0xca, // Array de elementos de 4 bytes
    sint16_array = 0xcb, // Array de elementos de 2 bytes
    sint32_array = 0xcc, // Array de elementos de 4 bytes
    real32_array = 0xcd, // Array de elementos de 4 bytes
    ubyte_xy = 0xd0, //2 bytes
    uint16_xy = 0xd1, //4 bytes
    uint32_xy = 0xd2, //8 bytes
    sint16_xy = 0xd3, //4 bytes
    sint32_xy = 0xd4, //8 bytes
    real32_xy = 0xd5, //8 bytes
    ubyte_box = 0xe0, //4 bytes
    uint16_box = 0xe1, //8 bytes
    uint32_box = 0xe2, //16 bytes
    sint16_box = 0xe3, //8 bytes
    sint32_box = 0xe4, //16 bytes
    real32_box = 0xe5 //16 bytes
}

```

Detalle 7. Enumeración con tipos de datos PCLXL

- 👁 *Operadores y Atributos utilizan 1 byte. Para identificar el Tipo de Datos se utiliza 1 byte pero requiere de campos adicionales para identificar el valor asociado para Tipos de Datos compuestos.*
 - 👁 *La codificación para Operadores y Tipos de Datos es excluyente. Esto permite facilitar el análisis de un fichero con lenguaje PCLXL*
-

Se siguen las siguientes reglas para obtener los valores de los atributos según el Tipo de Dato utilizado:

Para **tipos de datos simples** (tamaño fijo): {ubyte | uint16 | uint32 | sint16 | sint32 | real32 | ubyte_xy | uint16_xy | uint32_xy | sint16_xy | sint32_xy | real32_xy | ubyte_box | uint16_box | uint32_box | sint16_box | sint32_box | real32_box}

Tipo de dato(1 byte): Indica el tipo de dato utilizado para guardar el valor.

Valor: siguientes bytes según el tamaño del tipo de dato.

💡 Por ejemplo la siguiente secuencia (en formato hexadecimal) contiene un tipo de dato simple y su valor: 0xC11205

Tipo Dato = 0xC1. El tipo de dato es un uint16, un entero sin signo de 2 bytes.

Valor = 0x1205 (Son los 2 siguientes bytes; uint16 ocupa 2 bytes) representa el número 0x0512 (número decimal 1298, Hexadecimal Little Endian: Bytes menos significativos primero)

Para **tipos de datos compuestos** (arrays): {ubyte_array | uint16_array | uint32_array | sint16_array | sint32_array | real32_array}

El tipo de dato para arrays está compuesto por 3 elementos, de la siguiente forma: (A B C)

| | '-> (tamaño según tipo dato de C). Indica tamaño del array.
| '-> (1 byte).tipo de dato utilizado para el elemento C(tamaño array).
'-> (1 byte). Indica uno de los Tipo de dato array.

💡 Por ejemplo la secuencia: 0xC8C0021509 (Indica un array de 2 elementos de tipo ubyte.)

A: 0xC8 (1 byte): indica que es un array de tipos de datos ubyte.

B: 0xC0 (1 byte): Indica el tipo de datos del campo C necesario para guardar el tamaño del array. En este caso el tipo de datos es un ubyte.

C: 0x02(1 byte): Valor 2 en decimal. Quiere decir que el array consta de 2 elementos tipo ubyte.

Array = (0x15, 0x09) = (21, 9)

💡 Por ejemplo la secuencia: 0xC8C110010102030405011020.... (Se trata de un array de 272 elementos de tipo ubyte.)

A: 0xC8 (1 byte): indica que es un array de tipos de datos ubyte.

B: 0xC1 (1 byte): Indica el tipo de datos del campo C necesario para guardar el tamaño del array. En este caso el tipo de datos es un uint16.

C: 0x1001: Valor 272 en decimal (Hexadecimal Little Endian: Bytes menos significativos primero). Quiere decir que el array consta de 272 elementos de tipo ubyte.

Array[272]= (0x01, 0x02, 0x03, 0x04, 0x05, 0x01, 0x10, 0x20,) = (1, 2, 3, 4, 5, 1, 16, 32,)

2.2.5.3 EMF y EMFSPOOL

Toda la documentación sobre lenguaje EMF y EMFSPOOL está disponible en el MSDN de Microsoft como parte de la documentación de protocolos de Windows (Windows Protocols)(“Windows Protocols,” n.d.).

2.2.5.3.1 EMF (Enhanced Metafile Format)

Para una mejor comprensión del lenguaje EMF se puede consultar el manual [MS-EMF].pdf(“[MS-EMF]: Enhanced Metafile Format,” n.d.) en el MSDN de Microsoft.

El lenguaje EMF (Formato de Metarchivo Mejorado) consta de una colección de registros de tamaño variable que contienen comandos gráficos con los que se especifican operaciones de dibujo, objetos gráficos y propiedades con los que se puede representar una imagen gráfica. Las imágenes definidas con lenguaje EMF son independientes de cualquier dispositivo gráfico (impresora, plotter, pantalla, etc.) y mantienen su dimensión, forma y proporción.

Un metarchivo EMF consta de una secuencia de registros EMF. El primer registro es siempre un registro de cabecera (EMF Header) y el último es siempre un registro fin de fichero (EMF end-of-file record), entre estos dos registros se utilizan el resto de registros EMF para especificar las opciones de dibujo, configuración de propiedades y la creación de objetos gráficos.

Existen varias versiones de metarchivos EMF, estas son:

- **Original:** Primera versión que soporta comandos y objetos de dibujo independientes del dispositivo.
- **Extensión 1:** Añade un registro para definir el formato de pixel y añade soporte para comandos OpenGL, mejorando la independencia de los dispositivos y la flexibilidad de los metarchivos EMF.
- **Extensión 2:** La segunda extensión añade la capacidad de utilizar como unidad de medida de distancias micrómetros, mejorando la resolución y escalabilidad de las imágenes representadas por los metarchivos EMF.

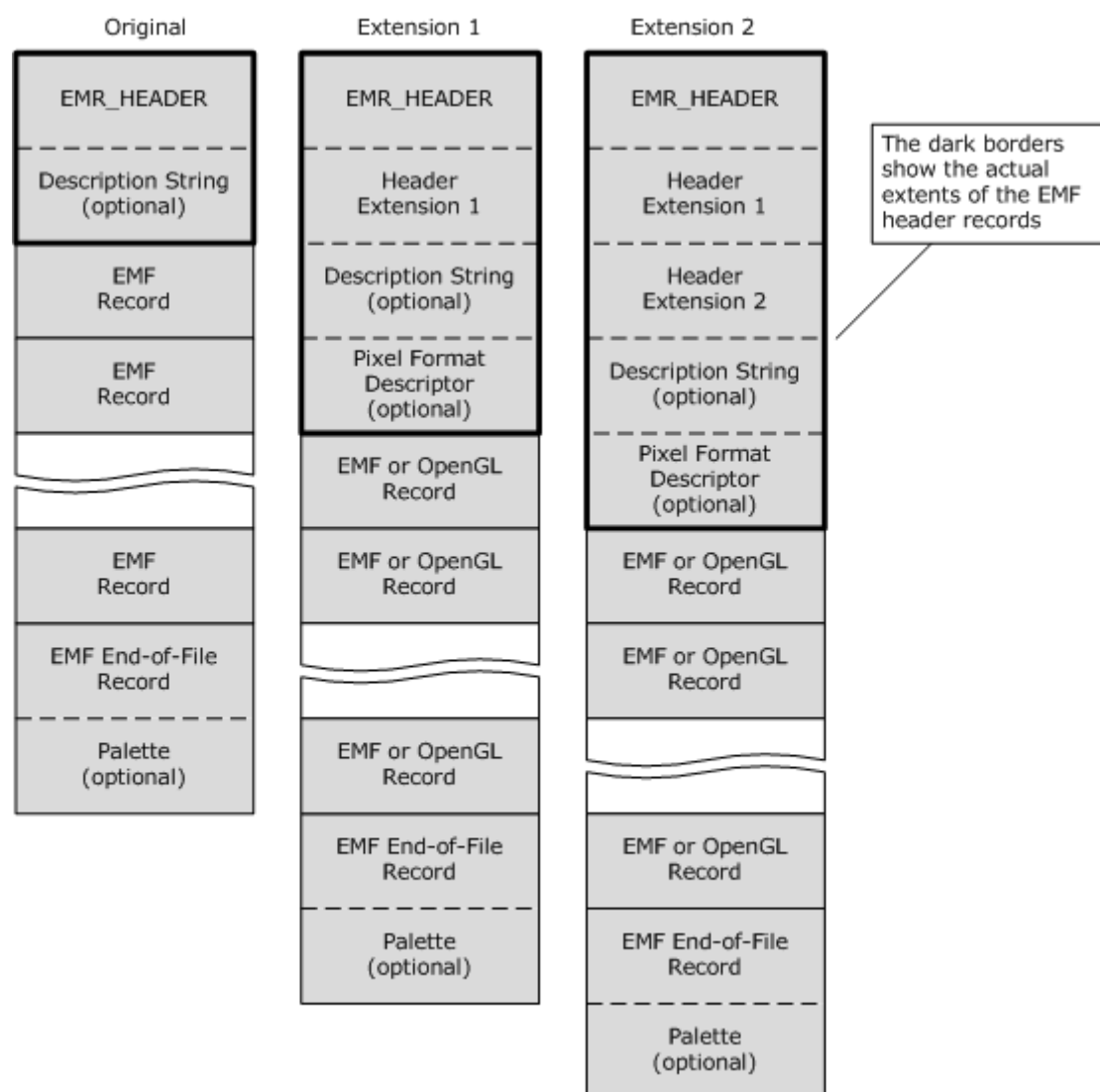


Ilustración 10 Imagen obtenida del documento MSDN: [MS-EMF].pdf

La ilustración pone de manifiesto, para cualquier versión EMF, que todos los metarchivos EMF tienen 3 secciones:

- **Cabecera EMF** (EMF Header): Contiene información de la estructura y contenido del metarchivo, incluye como información opcional una cadena de descripción (description string) y un descriptor de formato de pixel (Pixel Format Descriptor).
- **Registros EMF** (EMF records): Comprende un array de registros EMF que contienen toda la información necesaria para representar una imagen. Estos registros contienen comandos de dibujo, información de estado de los gráficos y definiciones de objetos gráficos. Al menos debe existir un registro, diferente al registro de cabecera EMF y al registro Fin-de-fichero EMF, si no existe, el metarchivo EMF no es válido.
- **Fin-de-fichero EMF** (EMF End-of-file): Señala el final de un metarchivo EMF. Tiene que ser el último registro de todos los registros EMF.

Dentro del registro de cabecera, la descripción y el descriptor de formato de pixel son opcionales y pueden situarse en cualquier orden, estos quedan determinados por los valores offset del registro de cabecera, si están presentes forman parte del registro de cabecera.

Los registros EMF son contiguos. Esto es necesario y significa que el metarchivo EMF se debe recorrer de forma secuencial porque la posición de cada registro EMF depende de la posición y tamaño del anterior registro EMF.

En cuanto a la estructura o campos que conforman los registros EMF, todos tienen una parte común, como se muestra en la siguiente tabla:

Type (4 bytes)
Size (4 bytes)
Resto de campos
.....
.....

- **Type:** Identifica el tipo de registro EMF. Este campo tiene una longitud fija de 4 bytes (entero de 32 bits sin signo).
- **Size:** Tamaño en bytes del registro EMF; comprende el tamaño de todos los campos incluidos los campos Type y Size. Este campo tiene una longitud fija de 4 bytes (entero de 32 bits sin signo).

👁 Con el contenido de estos dos campos se puede recorrer registro a registro el metarchivo EMF identificando todos los registros EMF que lo conforman. Así teniendo identificados los registros EMF podemos realizar un análisis selectivo de la información contenida en el metarchivo.

👁 En los ficheros de spool generados por el sistema de impresión se almacenan trabajos de impresión que pueden contener más de una página. Pero un metarchivo EMF sólo puede representar el contenido de una página. Por ello se necesita una ampliación de lenguaje EMF aplicado a los ficheros de spool: EMFSPOOL. Esto se verá en siguientes apartados. En el contexto de ficheros de spool podemos referirnos al lenguaje EMFSPOOL simplemente como lenguaje EMF.

2.2.5.3.1.1 Enumeración de registros EMF

La siguiente enumeración lista todos los registros EMF codificada por el campo Type:

```
enum EMR_RECORD
{
    EMR_HEADER = 0x00000001,
    EMR_POLYBEZIER = 0x00000002,
    EMR_POLYGON = 0x00000003,
    EMR_POLYLINE = 0x00000004,
    EMR_POLYBEZIERTO = 0x00000005,
    EMR_POLYLINETO = 0x00000006,
    EMR_POLYPOLYLINE = 0x00000007,
    EMR_POLYPOLYGON = 0x00000008,
    EMR_SETWINDOWEXTTEX = 0x00000009,
    EMR_SETWINDOWORGEX = 0x0000000A,
    EMR_SETVIEWPORTEXTTEX = 0x0000000B,
    EMR_SETVIEWPORTORGEX = 0x0000000C,
    EMR_SETBRUSHORGEX = 0x0000000D,
    EMR_EOF = 0x0000000E,
    EMR_SETPIXELV = 0x0000000F,
    EMR_SETMAPPERFLAGS = 0x00000010,
    EMR_SETMAPMODE = 0x00000011,
```

```
EMR_SETBKMODE = 0x00000012,  
EMR_SETPOLYFILLMODE = 0x00000013,  
EMR_SETROP2 = 0x00000014,  
EMR_SETSTRETCHBLTMODE = 0x00000015,  
EMR_SETTEXTALIGN = 0x00000016,  
EMR_SETCOLORADJUSTMENT = 0x00000017,  
EMR_SETTEXTCOLOR = 0x00000018,  
EMR_SETBKCOLOR = 0x00000019,  
EMR_OFFSETCLIPRGN = 0x0000001A,  
EMR_MOVETOEX = 0x0000001B,  
EMR_SETMETARGN = 0x0000001C,  
EMR_EXCLUDECLIPRECT = 0x0000001D,  
EMR_INTERSECTCLIPRECT = 0x0000001E,  
EMR_SCALEVIEWPORTEXTEX = 0x0000001F,  
EMR_SCALEWINDOWEXTEX = 0x00000020,  
EMR_SAVEDC = 0x00000021,  
EMR_RESTOREDC = 0x00000022,  
EMR_SETWORLDTRANSFORM = 0x00000023,  
EMR_MODIFYWORLDTRANSFORM = 0x00000024,  
EMR_SELECTOBJECT = 0x00000025,  
EMR_CREATEPEN = 0x00000026,  
EMR_CREATEBRUSHINDIRECT = 0x00000027,  
EMR_DELETEOBJECT = 0x00000028,  
EMR_ANGLEARC = 0x00000029,  
EMR_ELLIPSE = 0x0000002A,  
EMR_RECTANGLE = 0x0000002B,  
EMR_ROUNDRECT = 0x0000002C,  
EMR_ARC = 0x0000002D,  
EMR_CHORD = 0x0000002E,  
EMR_PIE = 0x0000002F,  
EMR_SELECTPALETTE = 0x00000030,  
EMR_CREATEPALETTE = 0x00000031,  
EMR_SETPALETTEENTRIES = 0x00000032,  
EMR_RESIZEPALETTE = 0x00000033,  
EMR_REALIZEPALETTE = 0x00000034,  
EMR_EXTFLOODFILL = 0x00000035,  
EMR_LINETO = 0x00000036,  
EMR_ARCTO = 0x00000037,  
EMR_POLYDRAW = 0x00000038,  
EMR_SETARCDIRECTION = 0x00000039,  
EMR_SETMITERLIMIT = 0x0000003A,  
EMR_BEGINPATH = 0x0000003B,  
EMR_ENDPATH = 0x0000003C,  
EMR_CLOSEFIGURE = 0x0000003D,  
EMR_FILLPATH = 0x0000003E,  
EMR_STROKEANDFILLPATH = 0x0000003F,  
EMR_STROKEPATH = 0x00000040,  
EMR_FLATTENPATH = 0x00000041,  
EMR_WIDENPATH = 0x00000042,  
EMR_SELECTCLIPPATH = 0x00000043,  
EMR_ABORTPATH = 0x00000044,  
EMR_COMMENT = 0x00000046,  
EMR_FILLRGN = 0x00000047,  
EMR_FRAMERGN = 0x00000048,  
EMR_INVERTRGN = 0x00000049,  
EMR_PAINTRGN = 0x0000004A,  
EMR_EXTSELECTCLIPRGN = 0x0000004B,  
EMR_BITBLT = 0x0000004C,  
EMR_STRETCHBLT = 0x0000004D,  
EMR_MASKBLT = 0x0000004E,  
EMR_PLGBLT = 0x0000004F,  
EMR_SETDIBITSTODEVICE = 0x00000050,
```

```

EMR_STRETCHDIBITS = 0x00000051,
EMR_EXTCREATEFONTINDIRECTW = 0x00000052,
EMR_EXTTEXTOUTA = 0x00000053,
EMR_EXTTEXTOUTW = 0x00000054,
EMR_POLYBEZIER16 = 0x00000055,
EMR_POLYGON16 = 0x00000056,
EMR_POLYLINE16 = 0x00000057,
EMR_POLYBEZIERTO16 = 0x00000058,
EMR_POLYLINETO16 = 0x00000059,
EMR_POLYPOLYLINE16 = 0x0000005A,
EMR_POLYPOLYGON16 = 0x0000005B,
EMR_POLYDRAW16 = 0x0000005C,
EMR_CREATEMONOBRUSH = 0x0000005D,
EMR_CREATEDIBPATTERNBRUSHPT = 0x0000005E,
EMR_EXTCREATEPEN = 0x0000005F,
EMR_POLYTEXTOUTA = 0x00000060,
EMR_POLYTEXTOUTW = 0x00000061,
EMR_SETICMMODE = 0x00000062,
EMR_CREATECOLORSPACE = 0x00000063,
EMR_SETCOLORSPACE = 0x00000064,
EMR_DELETECOLORSPACE = 0x00000065,
EMR_GLSRECORD = 0x00000066,
EMR_GLSBOUNDEDRECORD = 0x00000067,
EMR_PIXELFORMAT = 0x00000068,
EMR_DRAWESCAPE = 0x00000069,
EMR_EXTESCAPE = 0x0000006A,
EMR_SMALLTEXTOUT = 0x0000006C,
EMR_FORCEUFIMAPPING = 0x0000006D,
EMR_NAMEDESCAPE = 0x0000006E,
EMR_COLORCORRECTPALETTE = 0x0000006F,
EMR_SETICMPROFILEA = 0x00000070,
EMR_SETICMPROFILEW = 0x00000071,
EMR_ALPHABLEND = 0x00000072,
EMR_SETLAYOUT = 0x00000073,
EMR_TRANSPARENTBLT = 0x00000074,
EMR_GRADIENTFILL = 0x00000076,
EMR_SETLINKEDUFIS = 0x00000077,
EMR_SETTEXTJUSTIFICATION = 0x00000078,
EMR_COLORMATCHTOTARGETW = 0x00000079,
EMR_CREATECOLORSPACEW = 0x0000007A

```

};

Detalle 8. Enumeración EMR_RECORD. tipos de registros EMF

2.2.5.3.1.2 Tipos de registros EMF

Los registros EMF se pueden agrupar en las siguientes tipologías:

- Registros de mapas de bits (**Bitmap Record Types**): Con ellos se especifica cómo se representa un mapa de bits en el destino. Este tipo consta de los siguientes registros:

```

EMR_BITBLT
EMR_STRETCHBLT
EMR_MASKBLT
EMR_PLGBLT
EMR_SETDIBITSTODEVICE
EMR_STRETCHDIBITS
EMR_ALPHABLEND
EMR_TRANSPARENTBLT

```

Detalle 9. Registros EMF. Mapas de bits

- Registros de definición de regiones (**Clipping Record Types**): Se utiliza para definir y manipular regiones. Este tipo consta de los siguientes registros:

EMR_OFFSETCLIPRGN
 EMR_SETMETARGN
 EMR_EXCLUDECLIPRECT
 EMR_INTERSECTCLIPRECT
 EMR_SELECTCLIPPATH
 EMR_EXTSELECTCLIPRGN

Detalle 10. Registros EMF. Definición de regiones

- Registros comentario (**Comment record Types**): Se utilizan para incrustar otro tipo de contenido como registros EMFPLUS, EMFSPool, u otros tipo de contenido no EMF. Este tipo consta de los siguientes registros:

EMR_COMMENT_EMFSPool
 EMR_COMMENT_EMFPLUS
 EMR_COMMENT_PUBLIC

Detalle 11. Registros EMF. Comentarios

- Registros de Control (**Control record types**): Definen el comienzo y el fin de un metarchivo EMF y sus propiedades.

EMR_HEADER
 EMR_EOF

Detalle 12. Registros EMF. Control

Interesa ver el registro EMR_HEADER con más detalle, porque tiene información útil para el desarrollo del TFG:

Type (4 bytes)
Size (4 bytes)
EmfHeader (80 bytes)
.....
.....

En concreto el objeto incrustado EmfHeader con la siguiente estructura:

Bounds (16 bytes)
...
Frame (16 bytes)
...
RecordSignature (4 bytes)
Version (4 bytes)
Bytes (4 bytes)
Records (4 bytes)
Handles (2 bytes)
Reserved (2 bytes)
nDescription (4 bytes)
offDescription(4 bytes)
nPalEntries(4 bytes)
Device(8 bytes)

...
Millimeters (8 bytes)
...

👁 El campo **Frame** delimita la zona rectangular más pequeña que se puede dibujar alrededor de la imagen del metarchivo. Esta información es de utilidad para obtener el tamaño de papel en dispositivos de impresión tipo plotter.

- Registros para dibujar (**Drawing record types**): Definen operaciones de dibujo.

EMR_POLYBEZIER
EMR_POLYGON
EMR_POLYBEZIERTO
EMR_POLYLINETO
EMR_POLYPOLYLINE
EMR_POLYPOLYGON
EMR_SETPIXELV
EMR_ANGLEARC
EMR_ELLIPSE
EMR_RECTANGLE
EMR_ROUNDRECT
EMR_ARC
EMR_CHORD
EMR_PIE
EMR_EXTFLOODFILL
EMR_LINETO
EMR_ARCTO
EMR_POLYDRAW
EMR_FILLPATH
EMR_STROKEANDFILLPATH
EMR_STROKEPATH
EMR_FILLRGN
EMR_FRAMERGN
EMR_PAINTRGN
EMR_EXTTEXTOUTA
EMR_EXTTEXTOUTW
EMR_POLYBEZIER16
EMR_POLYGON16
EMR_POLYLINE16
EMR_POLYBEZIERTO16
EMR_POLYLINETO16
EMR_POLYPOLYLINE16
EMR_POLYPOLYGON16
EMR_POLYDRAW16
EMR_POLYTEXTOUTA
EMR_POLYTEXTOUTW
EMR_SMALLTEXTOUT
EMR_GRADIENTFILL

Detalle 13. Registros EMF. Dibujar

- Registros de secuencia de escape (**Escape record types**): Se utilizan para ejecutar funciones del driver de impresora.

EMR_DRAWESCAPE
EMR_EXTESCAPE
EMR_NAMEDESCAPE

Detalle 14. Registros EMF. Secuencia de escape

- Registros de creación de objetos (**Object creation record types**): Definen operaciones de creación de objetos gráficos.

EMR_CREATEMONOBRUSH
EMR_CREATEDIBPATTERNBRUSHPT
EMR_EXTCREATEPEN
EMR_CREATECOLORSPACEW
EMR_CREATEPEN
EMR_CREATEBRUSHINDIRECT
EMR_CREATEPALETTE
EMR_EXTCREATEFONTINDIRECTW
EMR_CREATECOLORSPACE

Detalle 15. Registros EMF. Creación de objetos

- Registros de manipulación de objetos (**Object manipulation record types**): Su función es administrar y modificar objetos gráficos.

EMR_SELECTOBJECT
EMR_DELETEOBJECT
EMR_SELECTPALETTE
EMR_SETPALETTEENTRIES
EMR_RESIZEPALETTE
EMR_SETCOLORSPACE
EMR_DELETECOLORSPACE
EMR_COLORCORRECTPALETTE

Detalle 16. Registros EMF. Manipulación de objetos

- Registros OpenGL (**OpenGL record types**): con ellos se especifica funciones OpenGL.

EMR_GLSRECORD
EMR_GLSBOUNDEDRECORD

Detalle 17. Registros EMF. OpenGL

- Registros de rutas de dibujo(**Path bracket record types**): Estos registros sirven para definir y manipular rutas de dibujo.

EMR_BEGINPATH
EMR_ENDPATH
EMR_CLOSEFIGURE
EMR_FLATTENPATH
EMR_WIDENPATH
EMR_ABORTPATH

Detalle 18. Registros EMF. Rutas de dibujo

- Registros de estado (**State record types**): Especifican y establecen las propiedades gráficas que define como se muestran los gráficos en los dispositivos gráficos.

EMR_SETBKCOLOR
EMR_SETWINDOWORGEX
EMR_SETVIEWPORTEXTEX
EMR_SETVIEWPORTORGEX
EMR_SETBRUSHORGEX
EMRSetColorAdjustment
EMR_MOVETOEX
EMR_SCALEVIEWPORTEXTEX
EMR_SETWINDOWEXTEX
EMR_SETMAPPERFLAGS
EMR_SETMAPMODE
EMR_SETBKMODE
EMR_SETPOLYFILLMODE
EMR_SETROP2
EMR_SETSTRETCHBLTMODE

```

EMR_SETTEXTALIGN
EMR_SETTEXTCOLOR
EMR_SETBKCOLOR
EMR_SCALEWINDOWEXT
EMR_SAVEDC
EMR_RESTOREDC
EMR_REALIZEPALETTE
EMR_SETARCDIRECTION
EMR_SETMITERLIMIT
EMR_INVERTGRN
EMR_SETICMMODE
EMR_PIXELFORMAT
EMR_FORCEUFIMAPPING
EMR_SETICMPROFILEA
EMR_SETICMPROFILEW
EMR_SETLAYOUT
EMR_SETLINKEDUFIS
EMR_SETTEXTJUSTIFICATION
EMR_COLORMATCHTOTARGETW

```

Detalle 19. Registros EMF. Estado

- Registros de transformación (**Transform record types**): define como se concreta el dibujo en la página física.

```

EMR_SETWORLDTRANSFORM
EMR_MODIFYWORLDTRANSFORM

```

Detalle 20. Registros EMF. Transformación

2.2.5.3.2 EMFSPOOL (Enhanced Metafile spool Format)

Para ampliar conocimiento sobre EMFSPOOL se puede consultar el manual [MS-EMFSPOOL].pdf(“[MS-EMFPLUS]: Enhanced Metafile Format Plus Extensions,” n.d.) en el MSDN de Microsoft.

Los trabajos de impresión almacenados en ficheros de spool pueden tener más de una página, pero un metarchivo EMF, con el conjunto de registros EMF disponibles, sólo puede almacenar el contenido de una página.

El lenguaje EMFSPOOL(Formato de Metarchivo de spool Mejorado) es una ampliación del lenguaje EMF donde se añaden nuevos registros de tipo EMFSPOOL y además de la capacidad del lenguaje EMF, nos permite representar el documento completo, incluyendo:

- La estructura general del documento.
- El formato y contenido de las páginas individuales.
- La configuración del dispositivo de impresión (como p.e. el tamaño del papel)
- Fuentes incrustadas en el propio documento.
- Imágenes de mapas de bits.
- Comandos PostScript inyectados.

La siguiente ilustración muestra la estructura de un fichero de spool tipo EMFSPOOL.

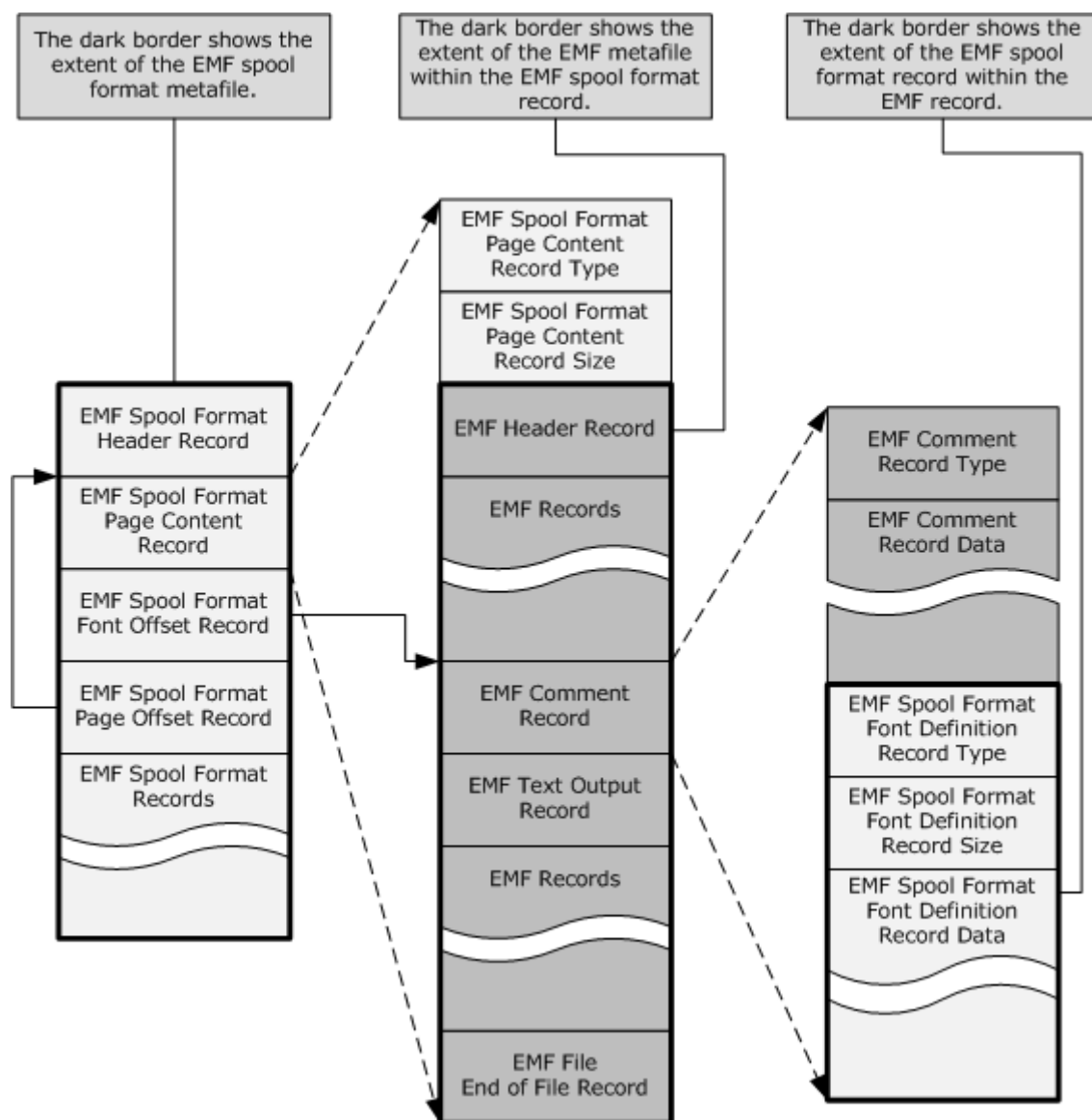


Ilustración 11 Imagen obtenida del documento MSDN: [MS-EMFSPPOOL].pdf

Como se aprecia en la ilustración un fichero con contenido EMFSPPOOL consta de:

- Un registro de cabecera. Siempre está presente y es el primer registro.
- Uno o varios registros de contenido de página, un registro por cada página del documento. Cada registro de contenido de página contiene un metarchivo EMF completo.
- La definición de una fuente se puede incrustar dentro de un registro EMF tipo EMR_COMMENT_EMFSPPOOL dentro de un metarchivo EMF que a su vez está dentro de un registro EMFSPPOOL de contenido de página.

En los siguientes apartados se describen los registros EMFSPPOOL.

2.2.5.3.2.1 Registro Cabecera (Header Record)

Es siempre el primer registro del metarchivo EMFSPPOOL. Además de indicar el inicio del metarchivo guarda información como el nombre del Documento y el dispositivo de salida. Tiene la siguiente estructura.

dwVersion(4 bytes)
cjSize (4 bytes)
dpszDocName(4 bytes)
dpszOutput(4 bytes)
.....
Extra.....

dwVersion: Identifica el registro de cabecera. Debe ser 0x00010000. Ocupa 4 bytes (entero de 32 bits sin signo).

cjSize: Tamaño total en bytes del registro de cabecera. Ocupa 4 bytes (entero de 32 bits sin signo).

dpszDocName: Posición dentro del registro de cabecera donde se encuentra el nombre del documento (este es una cadena Unicode terminada con un NULL). Es un campo de 4 bytes (entero de 32 bits sin signo), si su valor es 0x00000000 no se guarda el nombre del documento en el registro de cabecera.

dpszOutput: Posición dentro del registro de cabecera donde se encuentra el nombre del dispositivo de impresión, este es una cadena Unicode terminada con un NULL. Es un campo de 4 bytes (entero de 32 bits sin signo), si su valor es 0x00000000 no se guarda el nombre del dispositivo de impresión en el registro de cabecera.

👁 *El tamaño de todos los registros EMFSPOOL se redondean a múltiplos de 32 bits*

2.2.5.3.2.2 Registros de datos (Data Records)

El resto de registros EMFSPOOL se pueden clasificar como registros de datos con los que se representa el contenido del documento. Se pueden hacer las siguientes agrupaciones.

- **Registros con contenido de página** (Page Content records): Con ellos se especifica el contenido de una página.
- **Registros de posición de página** (Page Offset records): Especifica la posición del siguiente registro con contenido de página.
- **Registros de definición de fuentes** (Font Definition records): Especifica fuentes parciales, completas y sus propiedades.
- **Registros de posición de fuentes** (Font Offset records).
- **EMRI_DEVMODE.**
- **EMRI_PRESTARTPAGE.**
- **EMRI_PS_JOB_DATA.**

Todos ellos tienen la siguiente estructura genérica:

ulID(4 bytes)
cjSize (4 bytes)
data (variable)
.....
data

ulID: Identifica el tipo de registro. En la siguiente enumeración quedan recogidos todos los registros EMFSPOOL identificados por su ulID:

```
public enum EMRI_RECORD
{
    EMRI_EOF = 0x00000000,
    EMRI_METAFILE = 0x00000001,
    EMRI_ENGINE_FONT = 0x00000002,
    EMRI_DEVMODE = 0x00000003,
    EMRI_TYPE1_FONT = 0x00000004,
    EMRI_PRESTARTPAGE = 0x00000005,
    EMRI_DESIGNVECTOR = 0x00000006,
    EMRI_SUBSET_FONT = 0x00000007,
    EMRI_DELTA_FONT = 0x00000008,
    EMRI_FORM_METAFILE = 0x00000009,
    EMRI_BW_METAFILE = 0x0000000A,
    EMRI_BW_FORM_METAFILE = 0x0000000B,
    EMRI_METAFILE_DATA = 0x0000000C,
    EMRI_METAFILE_EXT = 0x0000000D,
    EMRI_BW_METAFILE_EXT = 0x0000000E,
    EMRI_ENGINE_FONT_EXT = 0x0000000F,
    EMRI_TYPE1_FONT_EXT = 0x00000010,
    EMRI_DESIGNVECTOR_EXT = 0x00000011,
    EMRI_SUBSET_FONT_EXT = 0x00000012,
    EMRI_DELTA_FONT_EXT = 0x00000013,
    EMRI_PS_JOB_DATA = 0x00000014,
    EMRI_EMBED_FONT_EXT = 0x00000015
};
```

Detalle 21. Registros EMFSPOOL.

cjSize: Indica el tamaño de los datos adjuntos al registro. Tamaño en bytes del resto del contenido del registro.

data: Es un array de tamaño variable, compuesto por una serie de campos, que contienen la información correspondiente al tipo de registro EMFSPOOL.

👁 *El tamaño en bytes de los registros de datos EMFSPOOL se obtiene de la suma del tamaño de los campos ulID + cjSize + tamaño de data indicado en el contenido del campo cjSize.*

2.2.5.3.2.3 Registro con contenido de página (Page Content records)

Consta de los siguientes registros:

-
- EMRI_METAFILE: La misma función que ENRI_METAFILE_DATA.
 - EMRI_FORM_METAFILE: La misma función que ENRI_METAFILE_DATA.
 - EMRI_BW_METAFILE: La misma función que ENRI_METAFILE_DATA. Excepto que el contenido es monocromo.
 - EMRI_BW_FORM_METAFILE: La misma función que ENRI_METAFILE_DATA. Excepto que el contenido es monocromo.
 - ENRI_METAFILE_DATA: Contiene un metarchivo EMF con el que se especifica el contenido de una página.
-

Detalle 22. Registros EMFSPOOL. Con contenido de página.

Todos tienen la misma estructura:

ulID(4 bytes)

cjSize (4 bytes)
EmfMetafile (variable)
.....
.....

EmfMetafile: Metarchivo EMF.

👁 Hay tantos registros de este tipo como páginas tiene el documento.

2.2.5.3.2.4 Registro de posición de página (Page Offset records)

Consta de los siguientes registros:

-
- EMRI_METAFILE_EXT: Indica la posición de un registro de contenido de página.
 - EMRI_BW_METAFILE_EXT: Igual que EMRI_METAFILE_EXT, pero el registro contenido de página sólo utiliza representación monocromo.
-

Detalle 23. Registros EMFSPool. De posición de página.

Sirven para indicar la posición del registro precedente de contenido de página. Tienen la siguiente estructura:

ulID(4 bytes)
cjSize (4 bytes)
offset (8 bytes: entero de 64 bits sin signo)

Offset: Contiene un entero sin signo de 64 bits que indica la distancia en bytes desde el anterior registro de contenido de página a la posición del actual registro de posición de página.

👁 Hay tantos registros de este tipo como páginas tiene el documento.

2.2.5.3.2.5 Registro de definición de fuentes (Font Definition records)

El contenido de estos registros se utiliza para definir fuentes. Consta de 5 registros distintos:

EMRI_ENGINE_FONT
EMRI_TYPE1_FONT
EMRI_DESIGNVECTOR
EMRI_SUBNET_FONT
EMRI_DELTA_FONT

Detalle 24. Registros EMFSPool. Definición de fuentes.

No se va a entrar en más detalle pues su estudio no es relevante para este trabajo.

2.2.5.3.2.6 Registro de posición de fuentes (Font Offset records)

Su función es determinar la posición de los registros de definición de fuentes. Consta de los siguientes 6 registros:

EMRI_ENGINE_FONT_EXT
EMRI_TYPE1_FONT_EXT
EMRI_DESIGNVECTOR_EXT
EMRI_SUBNET_FONT_EXT
EMRI_DELTA_FONT_EXT
EMRI_EMBED_FONT_EXT

Detalle 25. Registros EMFSPool. Posición de fuentes.

No se va a entrar en más detalle pues su estudio no es relevante para el trabajo que estamos realizando.

2.2.5.3.2.7 EMRI_DEVMODE record

Con este registro se especifica la configuración y capacidades del dispositivo de impresión. El registro tiene la siguiente estructura:

ulID(4 bytes)
cjSize (4 bytes)
Devmode (variable)

Devmode: Contiene una estructura _DEVMODE con la configuración y capacidades del dispositivo de impresión. Para más detalles consultar (“[MS-RPRN]: Print System Remote Protocol,” n.d.) del MSDN de Microsoft.

```
struct _DEVMODE
{
    DeviceName (64 bytes)
    SpecVersion (2 bytes)
    DriverVersion (2 bytes)
    Size (2 bytes)
    DriverExtra (2 bytes)
    Fields (4 bytes)
    Orientation (2 bytes)
    PaperSize (2 bytes)
    PaperLength (2 bytes)
    PaperWidth (2 bytes)
    Scale (2 bytes)
    Copies (2 bytes)
    DefaultSource (2 bytes)
    PrintQuality (2 bytes)
    Color (2 bytes)
    Duplex (2 bytes)
    YResolution (2 bytes)
    TTOption (2 bytes)
    Collate (2 bytes)
    FormName (64 bytes)
    Reserved0 (2 bytes)
    Reserved1 (4 bytes)
    Reserved2 (4 bytes)
    Reserved3 (4 bytes)
    Nup (4 bytes)
    Reserved4 (4 bytes)
    ICMMethod (4 bytes)
    ICMIntent (4 bytes)
    MediaType (4 bytes)
    DitherType (4 bytes)
    Reserved5 (4 bytes)
    Reserved6 (4 bytes)
    Reserved7 (4 bytes)
    Reserved8 (4 bytes)
    DriverExtraData (variable)
}
```

Detalle 26. Registros EMFSPool. _DEVMODE.

2.2.5.3.2.8 EMRI_PRESTARTPAGE record

Sirve para indicar el comienzo de PostScript Encapsulado.

No se va a entrar en más detalle pues su estudio no es relevante para el desarrollo del TFG.

2.2.5.3.2.9 EMRI_PS_JOB_DATA

Estos registros encapsulan datos PostScript a nivel de documento. Si se utiliza este registro debe aparecer inmediatamente después del registro de cabecera.

2.2.5.4 Ejemplo de análisis de ficheros de spool

Para finalizar con el análisis de ficheros de spool se muestra, en las siguientes dos ilustraciones, el contenido de un documento (Word) con 2 páginas.

A continuación se muestra el contenido de los ficheros de spool generados al imprimirse el documento por en una impresora HP LaserJet 5500 (utiliza lenguaje JPL/PCLXL) y por una impresora HP Laserjet 2025 (utilila lenguaje EMFSPPOOL). A continuación se muestra el análisis de los ficheros de spool según el lenguaje empleado por cada impresora.



Ilustración 12 Ejemplo para análisis de fichero de spool. Contenido primera página



Ilustración 13 Ejemplo para análisis de fichero de spool. Contenido segunda página

2.2.5.4.1 JPL/PCLXL

2.2.5.4.1.1 Contenido fichero de Spool

Cada posición de memoria representa el contenido de un byte en hexadecimal. La columna de la derecha la representación en formato ASCII si es un carácter.

Offset hex.	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	
000000000:	1B	25	2D	31	32	33	34	35	58	40	50	4A	4C	20	4A	4F	42	20	4E	41	4D	45	3D	22	._%-12345X@PJL JOB NAME="
000000018:	4D	69	63	72	6F	73	6F	66	74	20	57	6F	72	64	20	2D	20	45	6A	65	6D	70	6C	6F	Microsoft Word - Ejemplo
000000030:	2E	64	6F	63	78	22	0A	40	50	4A	4C	20	53	45	54	20	53	54	52	49	4E	47	43	4F	.docx".@PJL SET STRINGCO
000000048:	44	45	53	45	54	3D	55	54	46	38	0A	40	50	4A	4C	20	43	4F	4D	4D	45	4E	54	20	DESET=UTF8.@PJL COMMENT
000000060:	22	48	50	20	4C	61	73	65	72	4A	65	74	20	35	35	30	30	20	50	43	4C	20	36	2E	"HP LaserJet 5500 PCL 6.
000000078:	20	55	6E	69	76	65	72	73	61	6C	44	72	69	76	65	72	20	28	30	2E	33	2E	31	35	UniversalDriver (0.3.15
000000090:	38	34	2E	31	38	38	34	39	29	3B	20	57	69	6E	64	6F	77	73	20	37	20	45	6E	74	84.18849); Windows 7 Ent
0000000A8:	65	72	70	72	69	73	65	20	36	2E	31	2E	37	36	30	31	2E	31	3B	20	55	6E	69	64	erprise 6.1.7601.1; Unid
0000000C0:	72	76	20	30	2E	33	2E	37	36	30	31	2E	32	31	38	35	33	22	0A	40	50	4A	4C	20	rv 0.3.7601.21853".@PJL
0000000D8:	43	4F	4D	4D	45	4E	54	20	22	55	73	65	72	6E	61	6D	65	3A	20	6D	69	67	75	65	COMMENT "Username: migue
0000000F0:	6C	3B	20	41	70	70	20	46	69	6C	65	6E	61	6D	65	3A	20	4D	69	63	72	6F	73	6F	l; App Filename: Microso
000000108:	66	74	20	57	6F	72	64	20	2D	20	45	6A	65	6D	70	6C	6F	2E	64	6F	63	78	3B	20	ft Word - Ejemplo.docx;
000000120:	35	2D	31	38	2D	32	30	31	36	22	0A	40	50	4A	4C	20	43	4F	4D	4D	45	4E	54	20	5-18-2016".@PJL COMMENT
000000138:	22	4E	55	50	20	3D	20	4E	55	50	5F	31	22	0D	0A	40	50	4A	4C	20	53	45	54	20	"NUP = NUP_1"..@PJL SET
000000150:	4A	4F	42	41	54	54	52	3D	22	4F	53	3D	57	69	6E	64	6F	77	73	22	0A	40	50	4A	JOBATTR="OS=Windows".@PJ
000000168:	4C	20	53	45	54	20	4A	4F	42	41	54	54	52	3D	22	4F	53	20	56	65	72	73	69	6F	L SET JOBATTR="OS Versio
000000180:	6E	3D	57	69	6E	64	6F	77	73	20	37	20	45	6E	74	65	72	70	72	69	73	65	20	36	n=Windows 7 Enterprise 6
000000198:	2E	31	2E	37	36	30	31	2E	31	22	0A	40	50	4A	4C	20	53	45	54	20	4A	4F	42	41	.1.7601.1".@PJL SET JOBA
0000001B0:	54	54	52	3D	22	52	65	6E	64	65	72	20	54	79	70	65	3D	55	50	44	22	0A	40	50	TTR="Render Type=UPD".@P
0000001C8:	4A	4C	20	53	45	54	20	4A	4F	42	41	54	54	52	3D	22	52	65	6E	64	65	72	20	4E	JL SET JOBATTR="Render N
0000001E0:	61	6D	65	3D	48	50	20	55	6E	69	76	65	72	73	61	6C	20	50	72	69	6E	74	69	6E	ame=HP Universal Printin
0000001F8:	67	20	50	43	4C	20	36	22	0A	40	50	4A	4C	20	53	45	54	20	4A	4F	42	41	54	54	g PCL 6".@PJL SET JOBATT
000000210:	52	3D	22	52	65	6E	64	65	72	20	56	65	72	73	69	6F	6E	3D	36	31	2E	31	37	35	R="Render Version=61.175
000000228:	2E	31	2E	31	38	38	34	39	22	0A	40	50	4A	4C	20	53	45	54	20	4A	4F	42	41	54	.1.18849".@PJL SET JOBAT
000000240:	54	52	3D	22	4A	6F	62	41	63	63	74	31	3D	6D	69	67	75	65	6C	22	0A	40	50	4A	TR="JobAcct1=miguel".@PJ
000000258:	4C	20	53	45	54	20	4A	4F	42	41	54	54	52	3D	22	4A	6F	62	41	63	63	74	32	3D	L SET JOBATTR="JobAcct2=
000000270:	44	55	4B	45	33	44	22	0A	40	50	4A	4C	20	53	45	54	20	4A	4F	42	41	54	54	52	DUKE3D".@PJL SET JOBAT
000000288:	3D	22	4A	6F	62	41	63	63	74	33	3D	57	55	41	4C	22	0A	40	50	4A	4C	20	53	45	="JobAcct3=WUAL".@PJL SE
0000002A0:	54	20	4A	4F	42	41	54	54	52	3D	22	4A	6F	62	41	63	63	74	34	3D	32	30	31	36	T JOBATTR="JobAcct4=2016
0000002B8:	30	35	31	38	31	33	31	35	32	39	22	0A	40	50	4A	4C	20	53	45	54	20	4A	4F	42	0518131529".@PJL SET JOB
0000002D0:	41	54	54	52	3D	22	4A	6F	62	41	63	63	74	35	3D	65	34	38	66	30	62	33	62	2D	ATTR="JobAcct5=e48f0b3b-
0000002E8:	66	35	63	61	2D	34	34	63	31	2D	62	63	63	61	2D	37	35	30	62	66	62	31	35	63	f5ca-44c1-bcca-750bfb15c
000000300:	31	63	35	22	0A	40	50	4A	4C	20	53	45	54	20	4A	4F	42	41	54	54	52	3D	22	4A	1c5".@PJL SET JOBATTR="J
000000318:	6F	62	41	63	63	74	36	3D	4D	69	63	72	6F	73	6F	66	74	20	57	6F	72	64	22	0A	obAcct6=Microsoft Word".
000000330:	40	50	4A	4C	20	53	45	54	20	4A	4F	42	41	54	54	52	3D	22	4A	6F	62	41	63	63	@PJL SET JOBATTR="JobAcc
000000348:	74	37	3D	57	49	4E	57	4F	52	44	2E	45	58	45	22	0A	40	50	4A	4C	20	53	45	54	t7=WINWORD.EXE".@PJL SET
000000360:	20	4A	4F	42	41	54	54	52	3D	22	4A	6F	62	41	63	63	74	38	3D	6D	69	67	75	65	JOBATTR="JobAcct8=migue
000000378:	6C	22	0A	40	50	4A	4C	20	53	45	54	20	4A	4F	42	41	54	54	52	3D	22	4A	6F	62	1".@PJL SET JOBATTR="Job
000000390:	41	63	63	74	39	3D	22	0A	40	50	4A	4C	20	53	45	54	20	52	45	54	3D	4F	46	46	Acct9="".@PJL SET RET=OFF
0000003A8:	0A	40	50	4A	4C	20	44	4D	49	4E	46	4F	20	41	53	43	49	49	48	45	58	3D	22	30	.@PJL DMINFO ASCIIHEX="0
0000003C0:	34	30	30	30	34	30	31	30	31	30	32	30	44	31	30	31	30	30	31	31	35	33	32	33	400040101020D10100115323
0000003D8:	30	33	31	33	36	33	30	33	35	33	31	33	38	33	31	33	31	33	31	33	35	33	32	33	03136303531383131315323
0000003F0:	39	22	0A	40	50	4A	4C	20	53	45	54	20	48	4F	4C	44	3D	4F	46	46	0A	40	50	4A	9".@PJL SET HOLD=OFF.@PJ
000000408:	4C	20	53	45	54	20	55	53	45	52	4E	41	4D	45	3D	22	6D	69	67	75	65	6C	22	0A	L SET USERNAME="miguel".
000000420:	40	50	4A	4C	20	53	45	54	20	4A	4F	42	4E	41	4D	45	3D	22	4D	69	63	72	6F	73	@PJL SET JOBNAME="Micros
000000438:	6F	66	74	20	57	6F	72	64	20	2D	20	45	6A	65	6D	70	6C	6F	2E	64	6F	63	78	22	oft Word - Ejemplo.docx"
000000450:	0A	40	50	4A	4C	20	53	45	54	20	53	45	50	41	52	41	54	4F	52	50	41	47	45	3D	.@PJL SET SEPARATORPAGE=
000000468:	4F	46	46	0A	40	50	4A	4C	20	53	45	54	20	46	4F	4C	44	3D	22	22	0A	40	50	4A	OFF.@PJL SET FOLD="".@PJ
000000480:	4C	20	53	45	54	20	50	55	4E	43	48	3D	4F	46	46	0A	40	50	4A	4C	20	53	45	54	L SET PUNCH=OFF.@PJL SET
000000498:	20	50	52	4F	43	45	53	53	49	4E	47	41	43	54	49	4F	4E	3D	41	50	50	45	4E	44	PROCESSINGACTION=APPEND
0000004B0:	0A	40	50	4A	4C	20	53	45	54	20	50	52	4F	43	45	53	53	49	4E	47	54	59	50	45	.@PJL SET PROCESSINGTYPE
0000004C8:	3D	22	50	55	4E	43	48	22	0A	40	50	4A	4C	20	53	45	54	20	50	52	4F	43	45	53	="PUNCH".@PJL SET PROCES
0000004E0:	53	49	4E	47	4F	50	54	49	4F	4E	3D	22	4E	4F	4E	45	22	0A	40	50	4A	4C	20	53	SINGOPTION="NONE".@PJL S
0000004F8:	45	54	20	50	52	4F	43	45	53	45	49	4E	47	42	4F	55	4E	44	41	52	59	3D	4D	4F	ET PROCESSINGBOUNDARY=MO
000000510:	50	59	0A	40	50	4A	4C	20	53	45	54	20	51	54	59	3D	31	0A	40	50	4A	4C	20	53	PY.@PJL SET QTY=1.@PJL S
000000528:	45	54	20	4F	55	54	42	49	4E	3D	41	55	54	4F	0A	40	50	4A	4C	20	53	45	54	20	ET OUTBIN=AUTO.@PJL SET
000000540:	50	52	4F	43	45	53	53	49	4E	47	41	43	54	49	4F	4E	3D	41	50	50	45	4E	44	0A	PROCESSINGACTION=APPEND.
000000558:	40	50	4A	4C	20	53	45	54	20	50	52	4F	43	45	53	53	49	4E	47	54	59	50	45	3D	@PJL SET PROCESSINGTYPE=
000000570:	22	53	54	41	50	4C	49	4E	47	22	0A	40	50	4A	4C	20	53	45	54	20	50	52	4F	43	"STAPLING".@PJL SET PROC
000000588:	45	53	53	49	4E	47	4F	50	54	49	4F	4E	3D	22	4E										

Página 76

000000C48: 00 B5 00 B6 00 B7 00 B8 00 B9 00 BA 00 BB 00 BC 00 BE 00 C0 00 C1 00 C2 .µ.¶.·.¸.¹.º.»¼.½.¾.À.Á.Â
000000C60: 00 C2 00 B9 00 BA 00 BB 00 BC 00 BD 00 BE 00 BF 00 C0 00 C1 00 C2 00 C4 .Á.¹.º.»¼.½.¾.¿.À.Á.Â.Ã
000000C78: 00 BA 00 BB 00 BE 00 BF 00 C0 00 C2 00 C3 00 C4 00 B9 00 BA 00 BF 00 C0 .º.»¼.¿.À.Á.Â.Ã.Ä.Å.º.¿.À
000000C90: 00 C2 00 C6 00 C7 00 C9 00 CA 00 CB 00 CC 00 CF 00 C5 00 C7 00 CE 00 C5 .Ä.Å.Ç.É.Ê.Ë.Ì.Í.Î.Ï.Ð.Ñ.º.¿.À
000000CA8: 00 C7 00 C8 00 CF 00 D0 00 D2 00 D3 00 D7 01 0D 00 D0 00 D1 00 E9 00 D0 .Ç.È.Ï.Ð.Ò.Ó.×...Ð.Ñ.é.Ð
000000CC0: 00 FD 00 D0 05 E4 00 0E 00 41 00 00 05 BA 00 9C 00 9C 00 5E 00 5E C1 1C .ý.Ð.ä...Ä...º.æ.æ.^º.Á.
000000CD8: 09 F8 A7 50 FA 1C 09 00 00 40 47 64 5F 5E 5D 5C 5B 5A 59 58 55 54 53 52 .øSPú....@Gd_`\\[ZYXUTSR
000000CF0: 51 50 4F 4E 4D 4C 4B 4A 49 48 47 46 45 44 43 42 41 40 3F 3E 3D 3C 3B 3A QPONMLKJIHGFEDCBA@?>=<;:
000000D08: 39 38 37 36 35 2F 2E 2D 2C 28 26 25 24 23 22 1F 18 14 11 10 0F 0D 0B 0A 98765/./.,(&\$\$#".....
000000D20: 09 08 07 06 05 04 03 02 01 00 2C 45 23 46 60 20 B0 26 60 B0 04 26 23 48E#F`º&º.º#H
000000D38: 48 2D 2C 45 23 46 23 61 20 B0 26 61 B0 04 26 23 48 48 2D 2C 45 23 46 60 H-,E#F#aº&aº.º#HH-,E#F`
000000D50: B0 20 61 20 B0 46 60 B0 04 26 23 48 48 2D 2C 45 23 46 23 61 B0 20 60 20 ºaºFº.º#HH-,E#F#aº`
000000D68: B0 26 61 B0 20 61 B0 04 26 23 48 48 2D 2C 45 23 46 60 B0 40 61 20 B0 66 º&aºaº.º#HH-,E#F`º@aºf
000000D80: 60 B0 04 26 23 48 48 2D 2C 45 23 46 23 61 B0 40 60 20 B0 26 61 B0 40 61 º.º#HH-,E#F#aº@º&aº@a
000000D98: B0 04 26 23 48 48 2D 2C 01 10 20 3C 00 3C 2D 2C 20 45 23 20 B0 CD 44 23 º.º#HH-,...<.-,E#ºÍD#
000000DB0: 20 B8 01 5A 51 58 23 20 B0 8D 44 23 59 20 B0 ED 51 58 23 20 B0 4D 44 23 .,ZQX#ºD#YºíQX#ºMD#
000000DC8: 59 20 B0 04 26 51 58 23 20 B0 0D 44 23 59 21 21 2D 2C 20 20 45 18 68 44 Yº.&QX#ºD#Y!!--,E.hD
000000DE0: 20 B0 01 60 20 45 B0 46 76 68 8A 45 60 44 2D 2C 01 B1 0B 0A 43 23 43 65 º.ºEºFvhŠE`D-,±.±.C#Ce
000000DF8: 0A 2D 2C 00 B1 0A 0B 43 23 43 0B 2D 2C 00 B0 28 23 70 B1 01 28 3E 01 B0 .-,±.±.C#C-.,º.(#p±.(>º
000000E10: 28 23 70 B1 02 28 45 3A B1 02 00 08 0D 2D 2C 20 45 B0 03 25 45 61 64 B0 (#p±.(E:±....-,Eº.ºEadº
000000E28: 50 51 58 45 44 1B 21 21 59 2D 2C 20 45 B0 00 43 60 44 2D 2C 01 B0 06 43 PQXED.!!Y-,Eº.C`D-,º.C
000000E40: B0 07 43 65 0A 2D 2C 20 69 B0 40 61 B0 00 8B 20 B1 2C C0 8A 8C B8 10 00 º.Ce-,iº@aº.<±ÄŠE..
000000E58: 62 60 2B 0C 64 23 64 61 5C 58 B0 03 61 59 2D 2C 8A 03 45 8A 8A 87 B0 11 b`+.d#da\Xº.aY-,Š.EŠŠ+º.
000000E70: 2B B0 29 23 44 B0 29 7A E4 18 2D 2C 45 65 B0 2C 23 44 45 B0 2B 23 44 2D +º)#Dº)zä.-,Eeº,#DEº+#D-
000000E88: 2C 4B 52 58 45 44 1B 21 21 59 2D 2C 01 B0 05 25 10 23 20 8A F5 00 B0 01 61 23 ED EC 2D 2C ,KRXED.!!Y-,º.º.#Šöº.
000000EA0: 60 23 ED EC 2D 2C 01 B0 05 25 10 23 20 8A F5 00 B0 01 61 23 ED EC 2D 2C `#íì-,º.º.#Šöº.a#íì-,
000000EB8: 01 B0 06 25 10 F5 00 ED EC 2D 2C 20 B0 01 60 01 10 20 3C 00 3C 2D 2C 20 .º.º.ö.íì-,º.º.º.<.-,
000000ED0: B0 01 61 01 10 20 3C 00 3C 2D 2C 00 B0 07 43 B0 06 43 0B 2D 2C 21 21 0C º.a...<.-,º.Cº.C-.,!!
000000EE8: 64 23 64 8B B8 40 00 62 2D 2C 21 B0 80 51 58 0C 64 23 64 8B B8 20 00 62 d#d<,@.b-,!ºEQX.d#d<,.b
000000F00: 1B B2 00 40 2F 2B 59 B0 02 60 2D 2C 21 B0 C0 51 58 0C 64 23 64 8B B8 15 .º.º./+Yº.º.-,!ºÀQX.d#d<..
000000F18: 55 62 1B B2 00 80 2F 2B 59 B0 02 60 2D 2C 0C 64 23 64 8B B8 40 00 62 60 Ub.º.e/+Yº.º.-,d#d<,@.b`
000000F30: 23 21 2D 2C 45 23 45 60 23 45 60 23 45 60 23 76 68 18 B0 80 62 20 2D 2C #!-,E#E`#E`#E`#vh.ºEb -,
000000F48: B0 04 26 B0 04 26 B0 04 25 B0 04 25 45 23 45 20 B0 03 26 60 62 63 68 20 º.&º.&º.º.ºE#Eº.&bch
000000F60: B0 03 26 61 65 8A 23 44 44 2D 2C 20 45 B0 00 54 58 B0 40 44 20 45 B0 40 º.&aeŠDD-,Eº.TXº@D Eº@
000000F78: 61 44 1B 21 21 59 2D 2C 45 B1 30 2F 45 23 45 61 60 69 44 2D 2C aD.!!Y-,E±0/E#Eaº.º.ìD-,
000000F90: 4B 51 58 B0 2F 23 70 B0 14 23 42 1B 21 21 59 2D 2C 4B 51 58 20 B0 03 25 KQXº/#pº.#B.!!Y-,KQXº.º
000000FA8: 45 69 53 58 44 1B 21 21 59 1B 21 21 59 2D 2C 45 B0 14 43 B0 00 60 63 B0 EiSXD.!!Y.!!Y-,Eº.Cº.ºcº
000000FC0: 01 60 69 44 2D 2C B0 2F 45 44 2D 2C 45 23 20 45 8A 60 44 2D 2C 45 23 45 `ìD-,º/ED-,E# EŠ`D-,E#E
000000FD8: 60 44 2D 2C 4B 23 51 58 B9 00 33 FF E0 B1 34 20 1B B3 33 00 34 00 59 44 `D-,K#QX¹.3ÿà±4 .º.3.4.YD
000000FF0: 44 2D 2C B0 16 43 58 B0 03 26 45 8A 58 64 66 B0 1F 60 1B 64 B0 20 60 66 D-,º.CXº.&EŠXdfº.º.dº`f
000001008: 20 58 1B 21 B0 40 59 B0 01 61 59 23 58 65 59 B0 29 23 44 23 10 B0 29 E0 X.!!ºYº.aY#XeYº)#Dº.)à
000001020: 1B 21 21 21 21 21 59 2D 2C B0 16 43 58 B0 04 25 45 64 B0 20 60 66 20 58 .!!!!Y-,º.CXº.ºEdº`f X
000001038: 1B 21 B0 40 59 B0 01 61 23 58 65 59 B0 29 23 44 B0 04 25 B0 07 25 08 20 .!ºYº.a#XeYº)#Dº.º.º.
000001050: 58 02 1B 03 59 B0 05 25 10 B0 04 25 20 46 B0 04 25 23 42 3C B0 07 25 10 X...Yº.º.º.Fº.º#B<º.º.
000001068: B0 06 25 20 46 B0 04 25 B0 01 60 23 42 3C 20 58 01 1B 00 59 B0 05 25 10 º.º.Fº.º.º.`#B<X...Yº.º.
000001080: B0 04 25 B0 29 E0 B0 07 25 10 B0 06 25 B0 29 E0 B0 04 25 B0 07 25 08 20 º.º.)àº.º.º.º.º.º.º.º.
000001098: 58 02 1B 03 59 B0 04 25 B0 03 25 43 48 B0 06 25 B0 03 25 B0 01 60 43 48 X...Yº.º.º.CHº.º.º.`CH
0000010B0: 1B 21 59 21 21 21 21 21 2D 2C B0 16 43 58 B0 04 25 45 64 B0 20 60 .!Y!!!!!!-,º.CXº.ºEdº`
0000010C8: 66 20 58 1B 21 B0 40 59 B0 01 61 23 58 1B 65 59 B0 29 23 44 B0 05 25 B0 f X.!!ºYº.a#XeYº)#Dº.º.
0000010E0: 08 25 08 20 58 02 1B 03 59 B0 04 25 10 B0 05 25 20 46 B0 04 25 23 42 3C .º.X...Yº.º.º.Fº.º#B<
0000010F8: B0 04 25 B0 07 25 08 B0 07 25 10 B0 06 25 20 46 B0 04 25 B0 01 60 23 42 º.º.º.º.º.º.º.º.Fº.º.º.`#B
000001110: 3C 20 58 01 1B 00 59 B0 04 25 10 B0 05 25 B0 29 E0 B0 29 20 45 65 44 B0 <X...Yº.º.º.º.º.)àº) EeDº
000001128: 07 25 10 B0 06 25 B0 29 E0 B0 05 25 B0 08 25 08 20 58 02 1B 03 59 B0 05 .º.º.º.)àº.º.º.X...Yº.
000001140: 25 B0 03 25 43 48 B0 04 25 B0 07 25 08 B0 06 25 B0 03 25 B0 01 60 43 48 º.º.CHº.º.º.º.º.º.º.º.`CH
000001158: 1B 21 59 21 21 21 21 21 2D 2C 02 B0 04 25 20 20 46 B0 04 25 23 42 .!Y!!!!!!-,º.º.Fº.º#B
000001170: B0 05 25 08 B0 03 25 45 48 21 21 21 2D 2C 02 B0 03 25 20 B0 04 25 08 º.º.º.EH!!!!-,º.º.º.º.
000001188: B0 02 25 43 48 21 21 21 2D 2C 45 23 20 45 18 20 B0 00 50 20 58 23 65 23 º.CH!!!!-,E# E.º.P X#e#
0000011A0: 59 23 68 20 B0 40 50 58 21 B0 40 59 23 58 65 59 8A 60 44 2D 2C 4B 53 23 Y#hº@PX!º@Y#XeYŠ`D-,KS#
0000011B8: 4B 51 5A 58 20 45 8A 60 44 1B 21 21 59 2D 2C 4B 54 58 20 45 8A 60 44 1B KQZX EŠ`D.!!Y-,KTX EŠ`D.
0000011D0: 21 21 59 2D 2C 4B 53 23 4B 51 5A 58 38 1B 21 21 59 2D 2C B0 00 21 4B 54 !!Y-,KS#KQZX8.!!Y-,º.º!KT
0000011E8: 58 38 1B 21 21 59 2D 2C B0 02 43 54 58 B0 46 2B 1B 21 21 21 59 2D 2C X8.!!Y-,º.CTXºF+!!!!Y-,
000001200: B0 02 43 54 58 B0 47 2B 1B 21 21 59 2D 2C B0 02 43 54 58 B0 48 2B 1B º.CTXºG+!!!!Y-,º.CTXºH+.
000001218: 21 21 21 21 59 2D 2C B0 02 43 54 58 B0 49 2B 1B 21 21 21 59 2D 2C 20 8A !!!Y-,º.CTXºI+!!!!Y-,Š
000001230: 08 23 4B 53 8A 4B 51 5A 58 23 38 1B 21 21 59 2D 2C 00 20 B2 00 40 03 25 .#KSŠKQZX#8.!!Y-,º.º.º.
000001248: B0 06 26 49 61 8B 38 12 34 2D 2C 01 46 23 46 60 23 46 61 23 20 10 20 46 º.&Iaº.8.4-,F#F`#Fa# .F
000001260: 8A 61 B8 FF 80 62 8A B1 40 40 8A 70 45 60 68 3A 2D 2C 20 8A 23 49 64 8A Ša,ÿebŠ±@eŠpE`h:-,Š#IdŠ
000001278: 23 53 58 3C 1B 21 59 2D 2C 4B 52 58 7D 1B 7A 59 2D 2C B0 12 00 4B 01 4B #SX<.!Y-,KRX}.zY-,º.K.K.K
000001290: 54 42 2D 2C B1 02 00 42 B1 23 01 88 51 B1 40 01 88 53 5A 58 B9 10 00 00 TB-,±.B±#.ºQ±º.ºSZX¹...
0000012A8: 20 88 54 58 B2 02 01 02 43 60 42 59 B1 24 01 88 51 58 B9 20 00 00 40 88 ^TX²...C`B±\$º.QX¹ ..@^
0000012C0: 54 58 B2 02 02 02 43 60 42 B1 24 01 88 54 58 B2 02 20 02 43 60 42 00 4B TX²...C`B±\$º.TX². .C`B.K
0000012D8: 01 4B 52 58 B2 02 08 02 43 60 42 59 1B B9 40 00 00 80 88 54 58 B2 02 04 .KRX²...C`BY.º.º.e`TX²..

0000012F0: 02 43 60 42 59 B9 40 00 00 80 63 B8 01 00 88 54 58 B2 02 08 02 43 60 42 .C`BY¹@...€c...^TX²...C`B
000001308: 59 B9 40 00 01 00 63 B8 02 00 88 54 58 B2 02 10 02 43 60 42 59 B9 40 00 Y¹@...c...^TX²...C`BY¹@.
000001320: 02 00 63 B8 04 00 88 54 58 B2 02 40 02 43 60 42 59 59 59 59 2D 2C 45 .c...^TX².@.C`BYYY-,-E
000001338: 18 68 23 4B 51 58 23 20 45 20 64 B0 40 50 58 7C 59 68 8A 60 59 44 2D 2C .h#KQX# E d°@PX|YhŠ`YD-,-
000001350: B0 00 16 B0 02 25 B0 02 25 01 B0 01 23 3E 00 B0 02 23 3E B1 01 02 06 0C °...°.%°.%°.#>°.°.#>±....
000001368: B0 0A 23 65 42 B0 0B 23 42 01 B0 01 23 3F 00 B0 02 23 3F B1 01 02 06 0C °.°#eB°.°#B°.°#?°.°#>±....
000001380: B0 06 23 65 42 B0 07 23 42 B0 01 16 01 2D 2C 20 B8 20 00 62 8A 60 23 62 °.°#eB°.°#B°...,-, .bŠ`#b
000001398: 2D 2C B0 07 25 58 00 1B 01 59 B0 04 25 10 B0 03 25 B0 02 25 20 B8 FF FF -,°.%X...Y°.°.%°.%° .ýÿ
0000013B0: 54 58 21 CD 1B ED 59 21 B0 06 25 5C B8 69 59 B0 06 25 5A 58 B0 09 2B 59 20 B0 05 TX!í.íY!°.°%\°.°ZX°.°+Y°.°
0000013C8: 25 4A B0 04 25 47 B0 04 25 47 60 B0 06 25 47 B0 80 63 61 B0 02 25 B0 00 %J°.°G°.°G°.°G°Eca°.°%°.
0000013E0: 55 58 B0 03 25 B0 07 25 49 63 59 B0 08 25 58 00 1B 01 59 B0 04 25 B0 06 UX°.°%°.%IcY°.°%X...Y°.°%°.
0000013F8: 25 49 B0 09 25 5C B0 09 25 5A 58 B0 09 2B 59 B0 07 25 46 B0 80 63 61 B0 %I°.°%\°.°ZX°.°+Y°.°%F°Eca°
000001410: 03 25 20 B0 00 55 58 63 1B 21 59 61 23 20 B0 00 55 58 B0 80 63 1B 21 B0 .°.%°UXc.!Ya#°.°UX°€c.!°
000001428: 80 59 B0 59 2B B0 06 25 5C 58 69 59 B0 04 25 20 20 10 B0 00 48 23 3A B0 €Y°Y+°.°%\XiY°.° .°H#:.
000001440: 06 26 58 00 1B 01 59 B0 05 26 58 B0 03 25 2F 59 8A 12 23 32 21 21 2D 2C .&X...Y°.°&X°.°%/YŠ.#2!!-,-
000001458: B0 06 25 B0 0A 25 87 B0 06 25 B0 09 25 4A B0 00 53 58 B0 06 25 B0 0A 25 °.%°.%#°.°.%°%J°.°SX°.°%°.%
000001470: 1B B0 09 25 B0 07 25 59 B0 02 25 B0 02 25 07 0C B0 05 25 63 23 B0 06 25 °.%°.%Y°.°%°.%°...°%c#°.°%
000001488: 63 60 20 B9 40 00 04 00 63 53 58 21 B0 04 26 B0 04 26 B0 0A 1B B9 40 00 c`¹@...cSX!°.°&°.°¹@.
0000014A0: 04 00 63 65 51 58 B0 04 26 65 B0 04 26 65 B0 0A 1B B0 04 26 B0 0A 26 B0 ..ceQX°.°&e°.°&e°.°&°.°&°
0000014B8: 00 B0 03 25 B0 03 25 0B 0D 0A B0 09 2E B0 07 25 B0 07 25 0B 0D 0A B0 0B °.%°.%°...°.%°.%°...°.
0000014D0: 2E B0 05 25 B0 05 25 07 59 59 20 B0 00 55 58 B0 05 25 B0 05 25 87 B0 07 °.%°.%°Yÿ°.°UX°.°%°.%#°.°
0000014E8: 25 B0 07 25 0B B0 09 25 10 B0 0B 25 B0 09 26 20 B8 FF FF 54 58 21 CD 1B %°.°.%°.%°.%°& .ýÿTX!í.
000001500: ED 59 B0 05 25 B0 05 25 07 B0 08 25 B0 0B 25 49 23 B0 06 25 B0 06 25 87 íY°.°%°.%°.%°.%°%I#°.°%°.%#
000001518: B0 0A 25 10 B0 0B 25 C1 59 20 B0 00 51 B8 00 52 23 78 B0 01 61 B0 02 25 °.%°.%°ÁY°.°Q..R#x°.°a°.°%
000001530: B0 07 25 B0 07 25 07 B0 0A 25 B0 0D 25 49 61 B0 80 62 B0 05 25 B0 05 25 °.%°.%°.%°.%°%Ia°Ch°.°%°.%
000001548: 0B B0 0A 25 23 38 B0 06 25 B0 06 25 87 B0 08 25 B0 08 25 0B B0 0A 25 10 °.%°%#8°.°%°.%#°.°%°.%°.%
000001560: B0 0B 25 C4 B0 06 25 B0 06 25 07 B0 09 25 B0 0C 25 49 B0 03 25 54 B8 FF °.%°Á°.°%°.%°.%°.%°%I°.°%T,ý
000001578: A7 23 79 21 21 21 21 21 21 21 21 21 21 21 21 21 2D 2C 23 B0 00 54 58 B9 40 \$#y!!!!!!!-,-,°#°.°TX¹@
000001590: 00 00 00 1B B9 00 00 40 00 59 8A B0 00 54 58 B9 40 00 00 1B B9 00 00¹.°@.YŠ°.°TX¹@....¹..
0000015A8: 40 00 59 B0 5B 2B 2D 2C 08 B0 00 54 58 B9 40 00 00 00 1B B9 00 40 00 @.Y°[+-,°.°&°.°TX¹@....¹.°@.
0000015C0: 59 0D B0 5B 2B 2D 2C 8A 8A 08 0D 8A B0 00 54 58 B9 40 00 00 00 1B B9 00 Y°.°[+-,ŠŠ..Š°.°TX¹@....¹.
0000015D8: 00 40 00 59 B0 5B 2B 2D 2C B0 04 26 B0 04 26 08 0D B0 04 26 B0 04 26 08 @.Y°[+-,°.°&°.°&°.°&°.°&.
0000015F0: 0D B0 5B 2B 2D 2C 20 45 69 44 2D 00 00 C1 38 00 F8 A7 50 FB 38 00 01 00 °°[+-, Eid-..Á8.øSPû8...
000001608: 00 00 05 BD 71 5D B6 EF 65 5F 0F 3C F5 00 19 08 00 00 00 00 BB EB 7C ...¼q|¶ie_<ð.....»è|
000001620: CC 00 00 00 00 D2 0F 78 99 FB FA FD 8B 09 EC 07 B6 00 00 00 09 00 02 00 ì....ð.x™ûúý<.ì.¶.....
000001638: 01 00 00 00 00 C1 20 00 F8 A7 50 FB 20 00 01 00 00 0F 29 01 20 00 10 00Ä.øSPû.....¶).....
000001650: C3 00 07 00 02 00 10 00 2F 00 65 00 00 11 92 24 14 00 04 00 01 C1 00 20 Ä...../e....'\$.Ä.
000001668: F8 A7 50 FA 00 20 00 00 B2 09 03 20 B8 02 84 B4 09 18 01 4A 20 B8 02 83 øSPû. ...².....„'....J .,f
000001680: B4 09 18 01 4A 20 B8 02 82 B4 09 18 01 4A 20 B8 02 81 B4 09 18 01 4A 20 '....J .,,'....J .,.'....J
000001698: B8 02 80 B4 09 18 01 4A 20 B8 02 7F B4 09 18 01 4A 20 B8 02 7E B4 09 18 ,e'....J .,□....J .,~....
0000016B0: 01 4A 20 B8 02 7D B4 09 18 01 4A 20 B8 02 7C B4 09 18 01 4A 20 B8 02 7B ,J .,,'....J .,|....J .,{
0000016C8: B4 09 18 01 4A 20 B8 02 7A B4 09 18 01 4A 20 B8 02 79 B4 09 18 01 4A 20 '....J .,z'....J .,y'....J
0000016E0: B8 02 78 B4 09 18 01 4A 20 B8 02 77 B4 09 18 01 4A 20 B8 02 76 B4 09 18 ,x'....J .,w'....J .,v'..
0000016F8: 01 4A 20 B8 02 75 B4 09 18 01 4A 20 B8 02 74 B4 09 18 01 4A 20 B8 02 73 ,J .,u'....J .,t'....J .,s
000001710: B4 09 18 01 4A 20 B8 02 71 B4 09 18 01 4A 20 B8 02 70 B4 09 18 01 4A 20 '....J .,q'....J .,p'....J
000001728: B8 02 6F B4 09 18 01 4A 20 B8 02 6E B4 09 18 01 4A 20 B8 02 6D B4 09 18 ,o'....J .,n'....J .,m'..
000001740: 01 4A 20 B8 02 6C B4 09 18 01 4A 20 B8 02 6B B4 09 18 01 4A 20 B8 02 6A ,J .,l'....J .,k'....J .,j
000001758: B4 09 18 01 4A 20 B8 02 69 B4 09 18 01 4A 20 B8 02 68 B4 09 18 01 4A 20 '....J .,i'....J .,h'....J
000001770: B8 02 67 B4 09 18 01 4A 20 B8 02 66 B4 09 18 01 4A 20 B8 02 65 B4 09 18 ,g'....J .,f'....J .,e'..
000001788: 01 4A 20 B8 02 64 B4 09 18 01 4A 20 B8 02 63 B4 09 18 01 4A 20 B8 02 62 ,J .,d'....J .,c'....J .,b
0000017A0: B4 09 18 01 4A 20 B8 02 61 B4 09 18 01 4A 20 B8 02 60 B4 09 18 01 4A 20 '....J .,a'....J .,~....J
0000017B8: B8 02 5F B4 09 18 01 4A 20 B8 02 5E B4 09 18 01 4A 20 B8 02 5D B4 09 18 ,~....J .,^'....J .,]'..
0000017D0: 01 4A 20 B8 02 5C B4 09 18 01 4A 20 B8 02 5B B4 09 18 01 4A 20 B8 02 5A ,J .,\'....J .,[....J .,Z
0000017E8: B4 09 18 01 4A 20 B8 02 59 B4 09 18 01 4A 20 B8 02 58 B4 09 18 01 4A 20 '....J .,Y'....J .,X'....J
000001800: B8 02 57 B4 09 18 01 4A 20 B8 02 56 B4 09 18 01 4A 20 B8 02 55 B4 09 18 ,W'....J .,V'....J .,U'..
000001818: 01 4A 20 B8 02 54 B4 09 18 01 4A 20 B8 02 53 B4 09 18 01 4A 20 B8 02 52 ,J .,T'....J .,S'....J .,R
000001830: B4 09 18 01 4A 20 B8 02 51 B4 09 18 01 4A 20 B8 02 50 B4 09 18 01 4A 20 '....J .,Q'....J .,P'....J
000001848: B8 02 4F B4 09 18 01 4A 20 B8 02 4E B4 09 18 01 4A 20 B8 02 4D B4 09 18 ,O'....J .,N'....J .,M'..
000001860: 01 4A 20 B8 02 4C B4 09 18 01 4A 20 B8 02 4B B4 09 18 01 4A 20 B8 02 4A ,J .,L'....J .,K'....J .,J
000001878: B4 09 18 01 4A 20 B8 02 49 B4 09 18 01 4A 20 B8 02 48 B4 09 18 01 4A 20 '....J .,I'....J .,H'....J
000001890: B8 02 47 B4 09 18 01 4A 20 B8 02 46 B4 09 18 01 4A 20 B8 02 45 B4 09 18 ,G'....J .,F'....J .,E'..
0000018A8: 01 4A 20 B8 02 44 B4 09 18 01 4A 20 B8 02 43 B4 09 18 01 4A 20 B8 02 42 ,J .,D'....J .,C'....J .,B
0000018C0: B4 09 18 01 4A 20 B8 02 41 B4 09 18 01 4A 20 B8 02 40 B4 09 18 01 4A 20 '....J .,A'....J .,E'....J
0000018D8: B8 02 3F B4 09 18 01 4A 20 B8 02 3E B4 09 18 01 4A 20 B8 02 3D B4 09 18 ,?'.....J .,>'.....J .,='..
0000018F0: 01 4A 20 B8 02 3C B4 09 18 01 4A 20 B8 02 3B B4 09 18 01 4A 20 B8 02 3A ,J .,<'....J .,;'.....J .,:
000001908: B4 09 18 01 4A 20 B8 02 39 B4 09 18 01 4A 20 B8 02 38 B4 09 18 01 4A 20 '....J .,9'....J .,8'....J
000001920: B8 02 37 B4 09 18 01 4A 20 B8 02 36 B4 09 18 01 4A 20 B8 02 35 B4 09 18 ,7'....J .,6'....J .,5'..
000001938: 01 4A 20 B8 02 34 B4 09 18 01 4A 20 B8 02 33 B4 09 18 01 4A 20 B8 02 32 ,J .,4'....J .,3'....J .,2
000001950: B4 09 18 01 4A 20 B8 02 31 B4 09 18 01 4A 20 B8 02 30 B4 09 18 01 4A 20 '....J .,1'....J .,0'....J
000001968: B8 02 2F B4 09 18 01 4A 20 B8 02 2E B4 09 18 01 4A 20 B8 02 2D B4 09 18 ,./'.....J .,~'.....J .,-'..
000001980: 01 4A 20 B8 02 2C B4 09 18 01 4A 20 B8 02 2B B4 09 18 01 4A 20 B8 02 2A ,J .,,'....J .,+'....J .,*

000001998: B4 09 18 01 4A 20 B8 02 29 B4 09 18 01 4A 20 B8 02 28 B4 09 18 01 4A 20 '...J ..)'...J ..('...J
0000019B0: B8 02 27 B4 09 18 01 4A 20 B8 02 26 B4 09 18 01 4A 20 B8 02 25 B4 09 18 '...'...J ..&'...J ..%'...
0000019C8: 01 4A 20 B8 02 24 B4 09 18 01 4A 20 B8 02 23 B4 09 18 01 4A 20 B8 02 22 'J ..\$'...J ..#'...J .."
0000019E0: B4 09 18 01 4A 20 B8 02 21 B4 09 18 01 4A 20 B8 02 20 B4 09 18 01 4A 20 '...'...J ..!'...J ..'...J
0000019F8: B8 02 1F B4 09 18 01 4A 20 B8 02 1E B4 09 18 01 4A 20 B8 02 1D B4 09 18 '...'...J ..'...J ..'...J
000001A10: 01 4A 20 B8 02 1C B4 09 18 01 4A 20 B8 02 1B B4 09 18 01 4A 20 B8 02 1A 'J ..'...J ..'...J ..
000001A28: B4 09 18 01 4A 20 B8 02 19 B4 09 18 01 4A 20 B8 02 18 B4 09 18 01 4A 20 '...'...J ..'...J ..'...J
000001A40: B8 02 17 B4 09 18 01 4A 20 B8 02 16 B4 09 18 01 4A 20 B8 02 15 B4 09 18 '...'...J ..'...J ..'...J
000001A58: 01 4A 20 B8 02 14 B4 09 18 01 4A 20 B8 02 13 B4 09 18 01 4A 20 B8 02 12 'J ..'...J ..'...J ..
000001A70: B4 09 18 01 4A 20 B8 02 11 B4 09 18 01 4A 20 B8 02 10 B4 09 18 01 4A 20 '...'...J ..'...J ..'...J
000001A88: B8 02 0F B4 09 18 01 4A 20 B8 02 0E B4 09 18 01 4A 20 B8 02 0D B4 09 18 '...'...J ..'...J ..'...J
000001AA0: 01 4A 20 B8 02 0C B4 09 18 01 4A 20 B8 02 0B B4 09 18 01 4A 20 B8 02 0A 'J ..'...J ..'...J ..
000001AB8: B4 09 18 01 4A 20 B8 02 09 B4 09 18 01 4A 20 B8 02 08 B4 09 18 01 4A 20 '...'...J ..'...J ..'...J
000001AD0: B8 02 07 B4 09 18 01 4A 20 B8 02 06 B4 09 18 01 4A 20 B8 02 05 B4 09 18 '...'...J ..'...J ..'...J
000001AE8: 01 4A 20 B8 02 04 B4 09 18 01 4A 20 B8 02 03 B4 09 18 01 4A 20 B8 02 02 'J ..'...J ..'...J ..
000001B00: B4 09 18 01 4A 20 B8 02 01 B4 09 18 01 4A 20 B8 02 00 B4 09 18 01 4A 20 '...'...J ..'...J ..'...J
000001B18: B8 01 FF B4 09 18 01 4A 20 B8 01 FE B4 09 18 01 4A 20 B8 01 FD B4 09 18 'ÿ'...J ..b'...J ..ÿ'..
000001B30: 01 4A 20 B8 01 FC B4 09 18 01 4A 20 B8 01 FB B4 09 18 01 4A 20 B8 01 FA 'J ..ü'...J ..ü'...J ..ú
000001B48: B4 09 18 01 4A 20 B8 01 F9 B4 09 18 01 4A 20 B8 01 F8 B4 09 18 01 4A 20 '...'...J ..ü'...J ..ø'...J
000001B60: B8 01 F7 B4 09 18 01 4A 20 B8 01 F6 B4 09 18 01 4A 20 B8 01 F5 B4 09 18 '÷'...J ..ø'...J ..ø'...
000001B78: 01 4A 20 B8 01 F4 B4 09 18 01 4A 20 B8 01 F3 B4 09 18 01 4A 20 B8 01 F2 'J ..ø'...J ..ó'...J ..ò
000001B90: B4 09 18 01 4A 20 B8 01 F1 B4 09 18 01 4A 20 B8 01 F0 B4 09 18 01 4A 20 '...'...J ..ñ'...J ..ó'...J
000001BA8: B8 01 EF B4 09 18 01 4A 20 B8 01 EE B4 09 18 01 4A 20 B8 01 ED B4 09 18 'ï'...J ..î'...J ..í'...
000001BC0: 01 4A 20 B8 01 EC B4 09 18 01 4A 20 B8 01 EB B4 09 18 01 4A 20 B8 01 EA 'J ..ì'...J ..ë'...J ..è
000001BD8: B4 09 18 01 4A 20 B8 01 E9 B4 09 18 01 4A 20 B8 01 E8 B4 09 18 01 4A 20 '...'...J ..é'...J ..è'...J
000001BF0: B8 01 E7 B4 09 18 01 4A 20 B8 01 E6 B4 09 18 01 4A 20 B8 01 E5 B4 09 18 'ç'...J ..æ'...J ..å'...
000001C08: 01 4A 20 B8 01 E4 B4 09 18 01 4A 20 B8 01 E3 B4 09 18 01 4A 20 B8 01 E2 'J ..ä'...J ..ã'...J ..â
000001C20: B4 09 18 01 4A 20 B8 01 E1 B4 09 18 01 4A 20 B8 01 E0 B4 09 18 01 4A 20 '...'...J ..á'...J ..à'...J
000001C38: B8 01 DF B4 09 18 01 4A 20 B8 01 DE B4 09 18 01 4A 20 B8 01 DD B4 09 18 'ß'...J ..B'...J ..Ý'...
000001C50: 01 4A 20 B8 01 DC B4 09 18 01 4A 20 B8 01 DB B4 09 18 01 4A 20 B8 01 DA 'J ..Ü'...J ..Û'...J ..Ú
000001C68: B4 09 18 01 4A 20 B8 01 D9 B4 09 18 01 4A 20 B8 01 D8 B4 09 18 01 4A 20 '...'...J ..Ü'...J ..Ø'...J
000001C80: B8 01 D7 B4 09 18 01 4A 20 B8 01 D6 B4 09 18 01 4A 20 B8 01 D5 B4 09 18 '×'...J ..Ö'...J ..Ö'...
000001C98: 01 4A 20 B8 01 D4 B4 09 18 01 4A 20 B8 01 D3 B4 09 18 01 4A 20 B8 01 D2 'J ..ö'...J ..ó'...J ..ò
000001CB0: B4 09 18 01 4A 20 B8 01 D1 B4 09 18 01 4A 20 B8 01 D0 B4 09 18 01 4A 20 '...'...J ..ñ'...J ..b'...J
000001CC8: B8 01 CF B4 09 18 01 4A 20 B8 01 CE B4 09 18 01 4A 20 B8 01 CD B4 09 18 'ï'...J ..î'...J ..í'...
000001CE0: 01 4A 20 B8 01 CC B4 09 18 01 4A 20 B8 01 CB B4 09 18 01 4A 20 B8 01 CA 'J ..ì'...J ..ë'...J ..è
000001CF8: B4 09 18 01 4A 20 B8 01 C9 B4 09 18 01 4A 20 B8 01 C8 B4 09 18 01 4A 20 '...'...J ..é'...J ..è'...J
000001D10: B8 01 C7 B4 09 18 01 4A 20 B8 01 C6 B4 09 18 01 4A 20 B8 01 C5 B4 09 18 'Ç'...J ..E'...J ..Å'...
000001D28: 01 4A 20 B8 01 C4 B4 09 18 01 4A 20 B8 01 C3 B4 09 18 01 4A 20 B8 01 C2 'J ..Ä'...J ..Ã'...J ..Â
000001D40: B4 09 18 01 4A 20 B8 01 C1 B4 09 18 01 4A 20 B8 01 C0 B4 09 18 01 4A 20 '...'...J ..Ä'...J ..Ã'...J
000001D58: B8 01 BF B4 09 18 01 4A 20 B8 01 BE B4 09 18 01 4A 20 B8 01 BD B4 09 18 'ç'...J ..¼'...J ..¼'...
000001D70: 01 4A 20 B8 01 BC B4 09 18 01 4A 20 B8 01 BB B4 09 18 01 4A 20 B8 01 BA 'J ..¼'...J ..»'...J ..°
000001D88: B4 09 18 01 4A 20 B8 01 B9 B4 09 18 01 4A 20 B8 01 B8 B4 09 18 01 4A 20 '...'...J ..¹'...J ..'...J
000001DA0: B8 01 B7 B4 09 18 01 4A 20 B8 01 B6 B4 09 18 01 4A 20 B8 01 B5 B4 09 18 '...'...J ..ŧ'...J ..µ'...
000001DB8: 01 4A 20 B8 01 B4 B4 09 18 01 4A 20 B8 01 B3 B4 09 18 01 4A 20 B8 01 B2 'J ..³'...J ..³'...J ..²
000001DD0: B4 09 18 01 4A 20 B8 01 B1 B4 09 18 01 4A 20 B8 01 B0 B4 09 18 01 4A 20 '...'...J ..±'...J ..'...J
000001DE8: B8 01 AF B4 09 18 01 4A 20 B8 01 AE B4 09 18 01 4A 20 B8 01 AD B4 09 18 '¯'...J ..®'...J ..-...
000001E00: 01 4A 20 B8 01 AC B4 09 18 01 4A 20 B8 01 AB B4 09 18 01 4A 20 B8 01 AA 'J ..¬'...J ..«'...J ..ª
000001E18: B4 09 18 01 4A 20 B8 01 A9 B4 09 18 01 4A 20 B8 01 A8 B4 09 18 01 4A 20 '...'...J ..©'...J ..'...J
000001E30: B8 01 A7 B4 09 18 01 4A 20 B8 01 A6 B4 09 18 01 4A 20 B8 01 A5 B4 09 18 'Š'...J ..|'...J ..¥'...
000001E48: 01 4A 20 B8 01 A4 B4 09 18 01 4A 20 B8 01 A3 B4 09 18 01 4A 20 B8 01 A2 'J ..š'...J ..£'...J ..¢
000001E60: B4 09 18 01 4A 20 B8 01 A1 B4 09 18 01 4A 20 B8 01 A0 B4 09 18 01 4A 20 '...'...J ..|'...J ..'...J
000001E78: B8 01 9F B4 09 18 01 4A 20 B8 01 9E B4 09 18 01 4A 20 B8 01 9D B4 09 18 'ÿ'...J ..ž'...J ..•...
000001E90: 01 4A 20 B8 01 9C B4 09 18 01 4A 20 B8 01 9B B4 09 18 01 4A 20 B8 01 9A 'J ..œ'...J ..>...J ..š
000001EA8: B4 09 18 01 4A 20 B8 01 99 B4 09 18 01 4A 20 B8 01 98 B4 09 18 01 4A 20 '...'...J ..™'...J ..~...J
000001EC0: B8 01 97 B4 09 18 01 4A 20 B8 01 96 B4 09 18 01 4A 20 B8 01 95 B4 09 18 '¯'...J ..-...J ..•...
000001ED8: 01 4A 20 B8 01 94 B4 09 18 01 4A 20 B8 01 93 B4 09 18 01 4A 20 B8 01 92 'J .."''...J .."''...J ..'
000001EF0: B4 09 18 01 4A 20 B8 01 91 B4 09 18 01 4A 20 B8 01 90 B4 09 18 01 4A 20 '...'...J ..'...J ..'...J
000001F08: B8 01 8F B4 09 18 01 4A 20 B8 01 8E B4 09 18 01 4A 20 B8 01 8D B4 09 18 '...'...J ..Ž'...J ..•...
000001F20: 01 4A 20 B8 01 8C B4 09 18 01 4A 20 B8 01 8B B4 09 18 01 4A 20 B8 01 8A 'J ..œ'...J ..<...J ..š
000001F38: B4 09 18 01 4A 20 B8 01 89 B4 09 18 01 4A 20 B8 01 88 B4 09 18 01 4A 20 '...'...J ..%...J ..'...J
000001F50: B8 01 87 B4 09 18 01 4A 20 B8 01 86 B4 09 18 01 4A 20 B8 01 85 B4 09 18 'ž'...J ..†'...J ..™...
000001F68: 01 4A 20 B8 01 84 B4 09 18 01 4A 20 B8 01 83 B4 09 18 01 4A 20 B8 01 82 'J ..„'...J ..f'...J ..,
000001F80: B4 09 18 01 4A 20 B8 01 81 B4 09 18 01 4A 20 B8 01 80 B4 09 18 01 4A 20 '...'...J ..•'...J ..€'...J
000001F98: B8 01 7F B4 09 18 01 4A 20 B8 01 7E B4 09 18 01 4A 20 B8 01 7D B4 09 18 '□'...J ..~'...J ..}'...
000001FB0: 01 4A 20 B8 01 7C B4 09 18 01 4A 20 B8 01 7B B4 09 18 01 4A 20 B8 01 7A 'J ..|'...J ..{'...J ..z
000001FC8: B4 09 18 01 4A 20 B8 01 79 B4 09 18 01 4A 20 B8 01 78 B4 09 18 01 4A 20 '...'...J ..y'...J ..x'...J
000001FE0: B8 01 77 B4 09 18 01 4A 20 B8 01 76 B4 09 18 01 4A 20 B8 01 75 B4 09 18 'w'...J ..v'...J ..u'...
000001FF8: 01 4A 20 B8 01 74 B4 09 18 01 4A 20 B8 01 73 B4 09 18 01 4A 20 B8 01 72 'J ..t'...J ..s'...J ..r
000002010: B4 09 18 01 4A 20 B8 01 71 B4 09 18 01 4A 20 B8 01 70 B4 09 18 01 4A 20 '...'...J ..q'...J ..p'...J
000002028: B8 01 6F B4 09 18 01 4A 20 B8 01 6E B4 09 18 01 4A 20 B8 01 6D B4 09 18 '...'...J ..n'...J ..m...

000002040: 01 4A 20 B8 01 6C B4 09 18 01 4A 20 B8 01 6B B4 09 18 01 4A 20 B8 01 6A .J ,.l'...J ,.k'...J ,.j
000002058: B4 09 18 01 4A 20 B8 01 69 B4 09 18 01 4A 20 B8 01 68 B4 09 18 01 4A 20 '...J ,.i'...J ,.h'...J
000002070: B8 01 67 B4 09 18 01 4A 20 B8 01 66 B4 09 18 01 4A 20 B8 01 65 B4 09 18 .g'...J ,.f'...J ,.e'...
000002088: 01 4A 20 B8 01 64 B4 09 18 01 4A 20 B8 01 63 B4 09 18 01 4A 20 B8 01 62 .J ,.d'...J ,.c'...J ,.b
0000020A0: B4 09 18 01 4A 20 B8 01 61 B4 09 18 01 4A 20 B8 01 60 B4 09 18 01 4A 20 '...J ,.a'...J ,.'...J
0000020B8: B8 01 5F B4 09 18 01 4A 20 B8 01 5E B4 09 18 01 4A 20 B8 01 5D B4 09 18 .-...J ,.^'...J ,.]'...
0000020D0: 01 4A 20 B8 01 5C B4 09 18 01 4A 20 B8 01 5B B4 09 18 01 4A 20 B8 01 5A .J ,.\'...J ,.[...J ,.Z
0000020E8: B4 09 18 01 4A 20 B8 01 59 B4 09 18 01 4A 20 B8 01 58 B4 09 18 01 4A 20 '...J ,.Y'...J ,.X'...J
000002100: B8 01 57 B4 09 18 01 4A 20 B8 01 56 B4 09 18 01 4A 20 B8 01 55 B4 09 18 .W'...J ,.V'...J ,.U'...
000002118: 01 4A 20 B8 01 54 B4 09 18 01 4A 20 B8 01 53 B4 09 18 01 4A 20 B8 01 52 .J ,.T'...J ,.S'...J ,.R
000002130: B4 09 18 01 4A 20 B8 01 50 B4 09 18 01 4A 20 B8 01 4F B4 09 18 01 4A 20 '...J ,.P'...J ,.O'...J
000002148: B8 01 4E B4 09 18 01 4A 20 B8 01 4C B4 09 18 01 4A 20 B8 01 4B B4 09 18 .N'...J ,.L'...J ,.K'...
000002160: 01 4A 20 B8 01 4A B4 09 18 01 4A 20 B8 01 49 B4 09 18 01 4A 20 B8 01 48 .J ,.J'...J ,.I'...J ,.H
000002178: B4 09 18 01 4A 20 B8 01 47 B4 09 18 01 4A 20 B8 01 46 B4 09 18 01 4A 20 '...J ,.G'...J ,.F'...J
000002190: B8 01 45 B4 09 18 01 4A 20 B8 01 44 B4 09 18 01 4A 20 B8 01 43 B3 09 18 .E'...J ,.D'...J ,.C³..
0000021A8: 01 4A B8 02 84 B2 1E 3E 1F B8 02 83 B2 1E 15 1F B8 02 82 B2 1E 3E 1F B8 .J ,.².>.,.f²....,².>.,
0000021C0: 02 81 B2 1E 3A 1F B8 02 80 B2 1E 3E 1F B8 02 7F B2 1E 3E 1F B8 02 7E B2 .•².:.,.€².>.,.□².>.,.~²
0000021D8: 1E 0C 1F B8 02 7D B2 1E 3E 1F B8 02 7C B2 1E 3E 1F B8 02 7B B2 1E 3E 1F}².>.,.|².>.,.{².>..
0000021F0: B8 02 7A B2 1E 3E 1F B8 02 79 B2 1D 41 1F B8 02 78 B2 1D 43 1F B8 02 77 .z².>.,.y².A.,.x².C.,.w
000002208: B2 1D 43 1F B8 02 76 B2 1D 43 1F B8 02 75 B2 1D 41 1F B8 02 74 B2 1D 43 ².C.,.v².C.,.u².A.,.t².C
000002220: 1F B8 02 73 B2 1D 43 1F B8 02 72 B2 1D 41 1F B8 02 71 B2 1D 42 1F B8 02 .s².C.,.r².A.,.q².B.,.
000002238: 70 B2 1D 42 1F B8 02 6F B2 1D 42 1F B8 02 6E B2 1D 43 1F B8 02 6D B2 1D p².B.,.o².B.,.n².C.,.m²..
000002250: 43 1F B8 02 6C B2 1D 43 1F B8 02 6B B2 1C 45 1F B8 02 6A B2 1C 45 1F B8 C.,.l².C.,.k².E.,.j².E.,
000002268: 02 69 B2 1C 45 1F B8 02 68 B2 1C 45 1F B8 02 67 B2 1C 45 1F B8 02 66 B2 .i².E.,.h².E.,.g².E.,.f²
000002280: 1C 44 1F B8 02 65 B2 1C 45 1F B8 02 64 B2 1C 45 1F B8 02 63 B2 1C 45 1F .D.,.e².E.,.d².E.,.c².E..
000002298: B8 02 62 B2 1C 45 1F B8 02 61 B2 1C 45 1F B8 02 60 B2 1C 45 1F B8 02 5F .b².E.,.a².E.,.².E.,..
0000022B0: B2 1C 45 1F B8 02 5E B2 1C 44 1F B8 02 5D B2 1C 45 1F B8 02 5C B2 1C 45 ².E.,.².D.,.]².E.,.\².E
0000022C8: 1F B8 02 5B B2 1C 45 1F B8 02 5A B2 1C 45 1F B8 02 59 B2 1C 45 1F B8 02 .,[².E.,.Z².E.,.Y².E.,..
0000022E0: 58 B2 1C 45 1F B8 02 57 B2 1C 45 1F B8 02 56 B2 1C 45 1F B8 02 55 B2 1C X².E.,.W².E.,.V².E.,.U²..
0000022F8: 45 1F B8 02 54 B2 1C 45 1F B8 02 53 B2 1C 45 1F B8 02 52 B2 1C 45 1F B8 E.,.T².E.,.S².E.,.R².E.,
000002310: 02 51 B2 1C 45 1F B8 02 50 B2 1C 45 1F B8 02 4F B2 1C 45 1F B8 02 4E B2 .Q².E.,.P².E.,.O².E.,.N²..
000002328: 1C 45 1F B8 02 4D B2 1C 45 1F B8 02 4C B2 1C 45 1F B8 02 4B B2 1C 45 1F .E.,.M².E.,.L².E.,.K².E..
000002340: B8 02 4A B2 1B 49 1F B8 02 49 B2 1B 49 1F B8 02 48 B2 1B 4A 1F B8 02 47 .J².I.,.I².I.,.H².J.,.G
000002358: B2 1B 4A 1F B8 02 46 B2 1B 4B 1F B8 02 45 B2 1B 4C 1F B8 02 44 B2 1B 4C ².J.,.F².K.,.E².L.,.D².L
000002370: 1F B8 02 43 B2 1B 4C 1F B8 02 42 B2 1B 4B 1F B8 02 41 B2 1B 49 1F B8 02 .C².L.,.B².K.,.A².I.,..
000002388: 40 B2 1B 49 1F B8 02 3F B2 1B 4A 1F B8 02 3E B2 1B 4A 1F B8 02 3D B2 1B ².I.,.².J.,.>².J.,.²..
0000023A0: 4A 1F B8 02 3C B2 1B 4B 1F B8 02 3B B2 1B 4B 1F B8 02 3A B2 1B 4C 1F B8 J.,.<².K.,.;².K.,.:².L.,
0000023B8: 02 39 B2 1B 4C 1F B8 02 38 B2 1B 4C 1F B8 02 37 B2 1B 4C 1F B8 02 36 B2 .9².L.,.8².L.,.7².L.,.6²
0000023D0: 1B 4C 1F B8 02 35 B2 1B 49 1F B8 02 34 B2 1B 49 1F B8 02 33 B2 1B 4A 1F .L.,.5².I.,.4².I.,.3².J..
0000023E8: B8 02 32 B2 1B 4A 1F B8 02 31 B2 1B 4A 1F B8 02 30 B2 1B 4B 1F B8 02 2F .²².J.,.1².J.,.0².K.,./
000002400: B2 1B 4B 1F B8 02 2E B2 1B 4C 1F B8 02 2D B2 1B 4C 1F B8 02 2C B2 1B 4C ².K.,.².L.,.².L.,.².L..
000002418: 1F B8 02 2B B2 1B 4C 1F B8 02 2A B2 1B 4C 1F B8 02 29 B2 1B 4C 1F B8 02 .,+².L.,.*².L.,.)².L.,..
000002430: 28 B2 1B 4C 1F B8 02 27 B2 1B 49 1F B8 02 26 B2 1B 49 1F B8 02 25 B2 1B (².L.,.'².I.,.&².I.,.%²..
000002448: 4A 1F B8 02 24 B2 1B 4A 1F B8 02 23 B2 1B 4A 1F B8 02 22 B2 1B 4B 1F B8 J.,.\$².J.,.#².J.,."².K.,
000002460: 02 21 B2 1B 4B 1F B8 02 20 B2 1B 4C 1F B8 02 1F B2 1B 4C 1F B8 02 1E B2 .!².K.,.².L.,.².L.,.².L..
000002478: 1B 4C 1F B8 02 1D B2 1B 4C 1F B8 02 1C B2 1B 4C 1F B8 02 1B B2 1B 4C 1F .L.,.².L.,.².L.,.².L..
000002490: B8 02 1A B2 1A 4E 1F B8 02 19 B2 1A 4E 1F B8 02 18 B2 1A 4F 1F B8 02 17 .².N.,.².N.,.².N.,.².O..
0000024A8: B2 1A 4F 1F B8 02 16 B2 1A 4E 1F B8 02 15 B2 1A 4E 1F B8 02 14 B2 1A 4F ².O.,.².N.,.².N.,.².O
0000024C0: 1F B8 02 13 B2 1A 4F 1F B8 02 12 B2 1A 4F 1F B8 02 11 B2 1A 4E 1F B8 02 .².O.,.².O.,.².O.,.².N.,
0000024D8: 10 B2 1A 4E 1F B8 02 0F B2 1A 4F 1F B8 02 0E B2 1A 4F 1F B8 02 0D B2 1A .².N.,.².O.,.².O.,.².O..
0000024F0: 4F 1F B8 02 0C B2 1A 4E 1F B8 02 0B B2 1A 4E 1F B8 02 0A B2 1A 4E 1F B8 O.,.².N.,.².N.,.².N..
000002508: 02 09 B2 1A 4F 1F B8 02 08 B2 1A 4F 1F B8 02 07 B2 1A 4F 1F B8 02 06 B2 .².O.,.².O.,.².O.,.².O..
000002520: 19 50 1F B8 02 05 B2 19 51 1F B8 02 04 B2 19 51 1F B8 02 03 B2 19 50 1F .P.,.².Q.,.².Q.,.².P..
000002538: B8 02 02 B2 19 51 1F B8 02 01 B2 19 51 1F B8 02 00 B2 19 51 1F B8 01 FF .².Q.,.².Q.,.².Q.,.².Ÿ
000002550: B2 19 51 1F B8 01 FE B2 19 50 1F B8 01 FD B2 19 51 1F B8 01 FC B2 19 51 ².Q.,.².P.,.².Ÿ².Q.,.².ü².Q
000002568: 1F B8 01 FB B2 19 51 1F B8 01 FA B2 19 50 1F B8 01 F9 B2 19 51 1F B8 01 .,.û².Q.,.².ú².P.,.².ù².Q.,
000002580: F8 B2 19 51 1F B8 01 F7 B2 19 51 1F B8 01 F6 B2 19 51 1F B8 01 F5 B2 18 ø².Q.,.².÷².Q.,.².÷².Q.,.².ð²..
000002598: 54 1F B8 01 F4 B2 18 53 1F B8 01 F3 B2 18 54 1F B8 01 F2 B2 18 54 1F B8 T.,.ð².S.,.ó².T.,.ð².T.,
0000025B0: 01 F1 B2 18 54 1F B8 01 F0 B2 18 53 1F B8 01 EF B2 18 54 1F B8 01 EE B2 .ñ².T.,.ð².S.,.í².T.,.î²
0000025C8: 18 54 1F B8 01 ED B2 18 54 1F B8 01 EC B2 18 53 1F B8 01 EB B2 18 53 1F .T.,.í².T.,.î².S.,.ê².S..
0000025E0: B8 01 EA B2 18 54 1F B8 01 E9 B2 18 54 1F B8 01 E8 B2 18 54 1F B8 01 E7 .,.ê².T.,.é².T.,.è².T.,.ç
0000025F8: B2 18 54 1F B8 01 E6 B2 18 53 1F B8 01 E5 B2 18 54 1F B8 01 E4 B2 18 53 ².T.,.æ².S.,.ä².T.,.ä².S
000002610: 1F B8 01 E3 B2 18 54 1F B8 01 E2 B2 18 54 1F B8 01 E1 B2 18 54 1F B8 01 .,.ä².T.,.ä².T.,.ä².T.,.ä².T.,
000002628: E0 B2 18 53 1F B8 01 DF B2 18 53 1F B8 01 DE B2 18 54 1F B8 01 DD B2 18 à².S.,.ß².S.,.ß².T.,.Ý²..
000002640: 54 1F B8 01 DC B2 18 54 1F B8 01 DB B2 18 54 1F B8 01 DA B2 18 53 1F B8 T.,.Û².T.,.Û².T.,.Û².S.,
000002658: 01 D9 B2 18 54 1F B8 01 D8 B2 18 54 1F B8 01 D7 B2 17 57 1F B8 01 D6 B2 .Û².T.,.Ø².T.,.×².W.,.Ö²
000002670: 17 56 1F B8 01 D5 B2 17 56 1F B8 01 D4 B2 17 57 1F B8 01 D3 B2 17 57 1F .V.,.Ö².V.,.Ö².W.,.Ó².W..
000002688: B8 01 D2 B2 17 57 1F B8 01 D1 B2 17 57 1F B8 01 D0 B2 17 56 1F B8 01 CF .,.Ö².W.,.Ñ².W.,.Ð².V.,.İ
0000026A0: B2 17 56 1F B8 01 CE B2 17 57 1F B8 01 CD B2 17 57 1F B8 01 CC B2 17 57 ².V.,.î².W.,.î².W.,.î².W
0000026B8: 1F B8 01 CB B2 17 57 1F B8 01 CA B2 17 56 1F B8 01 C9 B2 17 56 1F B8 01 .,.Ë².W.,.Ê².V.,.Ê².V.,
0000026D0: C8 B2 17 57 1F B8 01 C7 B2 17 57 1F B8 01 C6 B2 17 57 1F B8 01 C5 B2 17 È².W.,.Ç².W.,.Æ².W.,.Å².

Monitorizar impresión de documentos en Windows

000003AE0: 0E 00 00 00 05 00 00 FE 73 03 D2 05 0E 00 03 00 07 00 39 00 4D 00 4E 00ps.ð.....9.M.N.
000003AF8: 5B BC 00 0D 01 52 00 16 00 3A 01 B5 B5 15 44 44 08 2D 07 BE 01 43 00 02 [¼...R...:µµ.DD.-.¾.C..
000003B10: 00 20 01 54 00 08 00 01 01 43 40 0F 04 02 0D B7 1A 3A 3F 23 B7 35 06 8D . .T.....C@.....?#·5.·
000003B28: 03 11 49 B8 01 2D B4 3F 07 8D 02 02 00 3F FD DE FD CE 2F FD DE ED 12 39 ...I...?·...ýþýí/ýbí.9
000003B40: 2F ED 01 2F DD E1 DE E1 10 F1 CE 11 39 2F 33 E1 D4 E1 31 30 01 11 21 11 /i./Ýáßá.ñí.9/3áÔá10...!
000003B58: 01 11 21 11 01 14 0E 02 07 07 14 06 23 22 2E 02 35 27 26 36 33 33 32 3E ...!.....#"...5'&6332>
000003B70: 02 35 34 26 23 22 0E 02 23 22 2E 02 35 34 36 37 3E 03 33 32 1E 02 03 14 .54&#"...#"...5467>.32....
000003B88: 0E 02 23 22 2E 02 35 34 3E 02 33 32 1E 02 01 03 D2 FC 5A 03 4F FD 06 02 ..#"...54>.32....ðüz.Oý..
000003BA0: 7E 23 3F 56 33 05 1E 1A 0F 14 0D 06 05 02 1C 18 12 2D 3C 25 10 53 5B 25 ~#?V3.....-<%S[
000003BB8: 3C 2E 1D 07 03 07 06 03 05 08 07 28 37 42 22 4B 6C 45 22 DA 07 11 1D 15 <.....(7B"K1E"Ú....
000003BD0: 16 1D 12 08 08 12 1D 16 15 1D 11 07 FD DB 05 0E FA F2 05 0E FB 43 04 6CýÛ...úð...ûC.l
000003BE8: FB 94 03 06 3B 5C 3F 24 03 98 0A 08 02 04 06 06 B8 1E 18 19 2A 3B 21 49 ú"...;?\$.~.....,*;!I
000003C00: 56 0F 13 10 04 0B 13 10 11 16 08 07 12 11 0C 28 44 5C FD 86 15 1C 12 08 V.....(D\ýþ+....
000003C18: 08 12 1C 15 15 1D 12 08 08 12 1D FD 4D 54 C8 C1 10 00 4D 53 20 50 43 4CýMTËÁ...MS PCL
000003C30: 58 4C 46 6F 6E 74 20 30 30 31 F8 A8 C5 00 00 B8 42 F8 A6 C1 02 00 F8 AA XLFont 001ø"Á...Bø!Á...øª
000003C48: 6F D5 00 00 80 3F 00 00 80 3F F8 A4 65 C8 C1 10 00 4D 53 20 50 43 4C 58 oÖ...E?..E?øªEËÁ...MS PCLX
000003C60: 4C 46 6F 6E 74 20 30 30 31 F8 A8 52 C1 01 00 F8 A2 C1 FC 00 F8 A3 53 FB LFont 001ø"RÁ...øçÁü.ø£Sû
000003C78: FC 01 01 00 FA 00 AC 04 22 00 57 00 02 00 AC FF FA 03 C7 05 0E 00 1C 00 ù...ú...-".W...-ýÛ.Ç.....
000003C90: 29 00 78 B1 24 06 B8 FF E8 B3 10 01 4D 06 B8 FF F0 B3 0F 01 4D 06 B8 FF).x±\$.ýð³..M..ýð³..M..ý
000003CA8: F0 B3 0D 01 4D 06 B8 FF F6 B3 0C 01 4D 06 B8 FF E8 B3 0B 01 4D 06 B8 02 ð³..M..ýð³..M..ýð³..M..
000003CC0: 11 40 17 11 1D 18 10 01 4D 1D 10 0F 01 4D 1D 10 0C 0D 01 4C 1D 20 0B 01 .@.....M.....M.....L. .
000003CD8: 4D 1D B8 02 32 B2 00 11 25 B8 01 0A B3 06 06 0C 23 B8 01 0B B3 15 51 0C M...2²...%...³...#...³..Q.
000003CF0: 55 00 3F 3F ED 12 39 2F ED 01 2F D4 E1 2B 2B 2B 2B 10 E1 2B 2B 2B 2B 2B U..??í.9/í./Ôá++++.á++++
000003D08: 32 31 30 01 14 0E 02 23 23 11 14 0E 02 23 22 2E 02 35 11 34 36 33 21 32 210....##...#"...5.463!2
000003D20: 1E 04 07 34 2E 02 23 11 33 32 3E 02 03 C7 40 77 AD 78 93 09 13 21 19 ...4...##.32>..Ç@w-x"...!
000003D38: 19 21 14 08 2A 1A 01 15 2A 4D 69 6D 4C 29 B5 3B 57 5D 2C 9F 9B 4E 67 46 .!.*...*MimLµ;WJ\Ý>NgF
000003D50: 24 03 93 61 9C 6E 3C FE 2E 08 0C 07 05 05 07 0C 08 04 AA 28 22 07 17 3F \$.“aen<p.....ª ("..?
000003D68: 5C 79 57 4F 6A 34 0E FD F9 28 47 63 00 C1 02 00 F8 A2 C1 FC 00 F8 A3 53 \ýW0j4.ýù(Gç.Á...øçÁü.ø£S
000003D80: FB FC 01 01 00 FA 00 99 02 CA 01 8C 00 01 00 99 FF FA 02 A7 03 CA 00 33 úü...ú...Û.Ê.E...™ýú.\$.Ê.3
000003D98: 00 63 B4 70 35 01 26 0F B8 FF F6 B3 0C 01 4D 0F B8 FF E8 B3 0B 01 4D 0F .c´p5.&..ýð³..M..ýð³..M.
000003DB0: B8 FF F0 B4 0F 10 01 4C 0F B8 FF F0 B3 0D 01 4D 0F B8 01 F8 40 12 00 40 ýð´...L..ýð³..M..ýð³..M..ø@...@
000003DC8: 13 01 4D 00 40 11 01 4D 00 20 0D 01 4D 00 1A 0A B8 01 25 40 09 2B 4A 26 ..M.@...M..M...%@..+J&
000003DE0: 0F 15 20 49 15 4C 00 3F 3F 12 39 39 3F ED 01 2F C4 2B 2B 2B E1 2B 2B 2B .. I.L.??..99?í./Á+++á+++
000003DF8: 2B 39 31 30 5D 01 14 0E 02 23 22 2E 02 23 22 0E 02 07 11 14 0E 02 23 22 +910]....#"...#".....#"
000003E10: 2E 02 35 11 34 3E 02 33 32 1E 02 15 15 3E 03 33 32 1E 06 02 A7 02 06 09 ...54>.32....>.32...\$.
000003E28: 08 08 17 1D 24 15 19 30 35 3A 23 08 12 20 19 18 20 13 07 06 12 1C 17 16\$.05:##.
000003E40: 1D 10 07 25 41 3A 39 1D 0D 21 24 1D 0C 04 03 01 03 62 16 1E 11 09 09 09 ...%A:9...!\$......b.....
000003E58: 08 14 2E 4C 37 FD B0 08 0B 08 04 04 08 0B 08 03 84 08 0B 09 03 03 09 0B ...L7ý°.....
000003E70: 08 83 36 44 27 0F 03 08 0A 0A 09 0E 1B 00 C1 03 00 F8 A2 C1 FC 00 F8 A3 .f6D'.....Á...øçÁü.ø£
000003E88: 53 FB FC 01 01 00 FA 00 85 01 D6 01 5D 00 02 00 85 FF FA 01 51 05 2F 00 Sûü...ú...Ö.]......ýú.Q./.
000003EA0: 15 00 21 00 8C B7 3F 23 4F 23 6F 23 03 16 B8 FF E8 B3 0D 01 4D 16 B8 FF ...!..E?·#O#o##..ýð³..M..ý
000003EB8: F6 B3 0C 01 4D 16 B8 FF E8 B3 13 01 4D 16 B8 FF E0 B3 10 01 4D 16 B8 FF ó³..M..ýð³..M..ýð³..M..ýð³..M..
000003ED0: F0 B3 0F 01 4D 16 B8 FF E8 B3 0B 01 4D 16 BB 02 71 00 1C 00 15 FF F0 B3 ð³..M..ýð³..M..»..q...ýð³
000003EE8: 0D 01 4D 15 B8 FF F6 B3 0C 01 4D 15 B8 FF E8 B3 0B 01 4D 15 B8 FF F0 B4 ..M..ýð³..M..ýð³..M..ýð´
000003F00: 0F 10 01 4C 15 B8 01 F8 B6 0A 1F 40 0F 01 4D 1F B8 01 3E B4 19 10 49 05 ...L...ø¶...@...M...>´..I.
000003F18: 4C 00 3F 3F D6 ED 2B 01 2F E1 2B 2B 2B 2B D4 E1 2B 2B 2B 2B 2B 2B 31 30 L.??ô!+./á++++ôá+++++10
000003F30: 5D 25 14 0E 02 23 22 2E 02 35 11 34 3E 02 33 32 1E 02 15 13 14 06 23 22]%...#"...5.4>.32....#"
000003F48: 26 35 34 36 33 32 16 01 3E 08 12 20 19 18 20 13 07 07 13 20 18 19 20 12 &54632..>..
000003F60: 08 13 2C 3B 3A 2B 2C 3B 3A 2B 19 08 0B 08 04 04 08 0B 08 03 84 07 0C 08 ...;+;+;+;+.....
000003F78: 04 04 08 0C 07 01 30 3A 2A 29 39 3A 2A 29 00 C1 04 00 F8 A2 C1 CC 01 F80:*)9:*)..Á...øçÁü.ø
000003F90: A3 53 FA CC 01 00 00 01 01 01 CA 00 99 06 64 01 75 00 01 00 99 FF FA 05 £SûÏ.....Ê...™d.u...™ýú.
000003FA8: D0 03 CA 00 50 00 EA 40 1B 12 10 0B 01 4D 13 18 0B 01 4D 26 10 0B 0C 01 ð.Ê.P.ê@.....M...M&...
000003FC0: 4C 27 18 0B 0C 01 4C 70 52 01 46 13 B8 FF F6 B3 0C 01 4D 13 B8 FF E8 B3 L'....LpR.F..ýð³..M..ýð³
000003FD8: 0B 01 4D 13 B8 FF F0 B3 10 01 4D 13 B8 FF F0 B3 0F 01 4D 13 B8 FF EA B3 ..M..ýð³..M..ýð³..M..ýð³
000003FF0: 0D 01 4D 13 B8 01 F8 40 1E 1E 16 0D 01 4D 1E 1E 32 0A 0A 0C 01 4D 0A 10 ..M..ø@.....M..2....M..
000004008: 0D 01 4D 0A 18 0B 01 4D 0A 10 0F 10 01 4C 0A B8 01 F9 B7 50 40 11 01 4D ..M...M...L...ù·P@..M
000004020: 50 3E 27 B8 FF F6 B3 0C 01 4D 27 B8 FF E8 B3 0B 01 4D 27 B8 FF F0 B3 10 P>'.ýð³..M'.ýð³..M'.ýð³..M'.ýð³.
000004038: 01 4D 27 B8 FF F0 B3 0F 01 4D 27 B8 FF F0 B3 0D 01 4D 27 BB 01 F8 00 32 .M'.ýð³..M'.ýð³..M'»..ø.2
000004050: 00 10 01 06 B2 4B 4A 24 B8 01 06 40 0D 41 4A 3E 46 2D 38 49 2D 4C 19 4C²KJ\$...@.AJ>F-8I-L.L
000004068: 05 4C 00 3F 3F 3F 3F 12 39 39 3F ED 3F ED 01 2F E1 2B 2B 2B 2B 2B 39 D4 .L.????.99?í?i./á+++++9Ô
000004080: 2B E1 2B 2B 2B 2B 11 39 2F 2B E1 2B 2B 2B 2B 2B 39 31 30 5D 00 2B 2B 2B +á++++.9/(á+++++910].+++
000004098: 2B 25 14 0E 02 23 22 2E 02 35 11 34 2E 02 23 22 06 07 11 14 0E 02 23 22 +%.#"...5.4...#".....#"
0000040B0: 2E 02 35 11 34 2E 02 23 22 06 07 11 14 0E 02 23 22 2E 02 35 11 34 3E 02 ...5.4...#".....#"...5.4>.
0000040C8: 33 32 1E 02 15 15 36 36 33 32 1E 02 17 3E 03 33 32 1E 02 15 05 D0 08 12 32....6632...>.32....ð..
0000040E0: 20 18 19 20 13 08 14 2C 44 2E 39 73 45 08 13 20 18 17 21 13 07 16 2C 43,D.9sE.. ...!...;C
0000040F8: 2E 39 74 44 08 12 20 19 18 20 13 07 06 12 1C 17 16 1D 10 07 4C 8F 49 38 .9tD..L·I8
000004110: 59 44 30 10 2D 51 4B 47 24 57 76 49 1F 19 08 0B 08 04 04 08 0B 08 02 23 YD0.-QKG\$WvI.....#
000004128: 39 5E 44 24 58 55 FD 8B 08 0B 08 04 04 08 0B 08 02 23 39 5E 44 24 58 55 9^D\$XUý<.....#9^D\$XU
000004140: FD 8B 08 0B 08 04 04 08 0B 08 03 84 08 0B 09 03 03 09 0B 08 77 55 4F 1A ý<.....wUO.
000004158: 2F 41 28 31 44 2A 13 3D 66 89 4C C1 05 00 F8 A2 C1 4C 01 F8 A3 53 FA 4C /A(1D*=f%LÁ...øçÁü.ø£SûL
000004170: 01 00 00 01 01 01 4A 00 5D 03 FB 01 1E 00 02 00 5D FF ED 03 99 03 CA 00J..].ù.....]ýi...Ê.

000004188: 2B 00 34 00 AE 40 2A 2F 36 3F 36 6F 36 03 13 20 0D 01 4D 13 20 0F 01 4D +.4.@@*/6?6o6...M. .M
0000041A0: 13 13 21 2C 10 0C 01 4D 2C 18 0B 01 4D 2C 10 0D 01 4D 2C 10 0F 10 01 4C !!,...M,...M,...L
0000041B8: 2C B8 01 F9 B5 44 2B A4 2B 02 2B B8 FF F0 B5 0F 01 4D 2B 34 04 B8 FF E0 ,.ùuD+#+.ÿöµ..M+4..ÿà
0000041D0: B3 0D 01 4D 04 B8 FF EE B3 0C 01 4D 04 B8 FF E0 B3 0B 01 4D 04 B8 FF E8 >..M..ÿi³..M..ÿà³..M..ÿè
0000041E8: B4 0F 10 01 4C 04 B8 02 0A 40 21 21 2C C7 80 04 90 04 02 C0 04 01 D0 04 ´...L...@!;,ÇE.●...Ä..D.
000004200: 01 04 04 1C 2F DC 26 4A 18 EA 60 0E 70 0E 02 0E 09 EA 1C 4D 00 3F FD D4/Ü&J.è`.p....è.M.ÿô
000004218: 5D ED 3F ED 12 39 2F 5D 71 72 ED 01 2F E1 2B 2B 2B 2B 32 DD 2B 5D E1 2B]i?i.9/]qri./á++++2ÿ+]á+
000004230: 2B 2B 2B 12 39 2F 2B 2B 31 30 5D 01 14 06 23 21 14 1E 02 33 32 3E 02 33 +++9/++10]...#!...32>.3
000004248: 32 1E 02 15 14 0E 06 23 22 2E 02 35 34 3E 02 33 32 1E 02 15 27 36 26 23 2.....#"..54>.32...!6&#
000004260: 22 0E 02 07 03 99 27 19 FD B2 1E 46 70 51 40 64 49 2F 0C 07 0B 06 03 02 "...."'.ÿ².FpQ@dI/.....
000004278: 03 07 0B 36 56 71 40 6F A7 72 3A 3C 71 A0 63 6A 95 60 2D A6 03 79 77 3D ...6Vq@o\$R:<q cj·`-!;yw=
000004290: 5C 3E 22 02 02 02 27 21 4B 78 56 2E 15 1A 15 07 0E 19 12 0D 13 0F 0C 0B \>"...!KxV.....
0000042A8: 19 18 13 3E 7A B8 7A 74 B9 81 45 44 73 9B 58 13 82 94 2E 4C 65 37 00 C1 ...>z,zt¹•EDsX,„.Le7.Á
0000042C0: 06 00 F8 A2 C1 68 01 F8 A3 53 FA 68 01 00 00 01 01 01 66 00 5C 03 D5 01 ...øçÄh.ø£Súh.....f.\.Ö.
0000042D8: 02 00 02 00 5C FF ED 03 3F 03 CA 00 33 00 40 00 B0 40 20 0F 42 1F 42 2F\ÿi.?.Ë.3.è.°@ .B.B/
0000042F0: 42 03 0B 09 17 34 0C 0C 01 4D 34 10 0D 01 4D 34 18 0B 01 4D 34 10 0F 10 B....4...M4...M4...M4...
000004308: 01 4C 34 B8 01 F7 40 0E 00 27 27 0F 00 01 0C 00 10 0F 01 4D 00 3A B8 FF .L4,÷@..''.....M.:;ÿ
000004320: F0 B3 0C 01 4D 3A B8 FF E8 B3 0D 01 4D 3A B8 FF E8 B3 0B 01 4D 3A B8 FF ô³..M.;ÿè³..M.;ÿè³..M.;ÿ
000004338: E8 B4 0F 10 01 4C 3A B8 02 08 40 14 11 17 C5 00 34 10 34 02 3B C0 34 01 è´...L:;..@...Ä.4.4.;Ä4.
000004350: 00 34 01 34 34 2E 41 2A B8 01 1B 40 0F D0 22 01 22 22 04 1D EA 2E 4A 3D .4.44.A*„.è.B"....."'.è.J=
000004368: DC 0C 4D 04 00 2F 3F ED 3F ED 12 39 2F 5D ED 11 12 39 2F 5D 71 5E 5D ED Ü.M../?i?i.9/]i..9/]q^]i
000004380: 01 2F E1 2B 2B 2B 2B C4 2B 5E 5D 39 2F 10 E1 2B 2B 2B 2B 32 39 31 30 5E ./á++++Ä+^]9/.á++++2910^
000004398: 5D 25 14 06 06 23 22 26 26 35 35 06 06 23 22 2E 02 35 34 3E 02 33 33 35]%...#"&55..#"..54>.335
0000043B0: 34 2E 02 23 22 0E 02 23 22 2E 02 35 34 3E 04 33 32 1E 02 15 07 23 22 0E 4..#"..#"..54>.32....#"..
0000043C8: 02 15 14 16 33 32 36 37 03 3F 10 1C 1B 1A 1F 0E 3B 91 51 47 73 51 2D 3E3267.?......;QGsq->
0000043E0: 74 A8 69 7C 16 31 4E 39 3D 61 49 31 0C 08 0C 09 04 07 1B 42 56 66 34 61 t`i|].1N9=aI1.....BVf4a
0000043F8: 88 54 26 A4 8D 44 64 42 1F 59 50 41 6F 3D 18 0C 0C 06 06 0C 0C 5A 3F 46 ^T&•DdB.YPAo=.....Z?F
000004410: 25 46 66 41 4C 70 4A 25 46 34 50 35 1B 1D 23 1D 08 10 19 0F 19 1D 1A 23 %FfALpJ%F4P5..#.....#
000004428: 1D 13 2C 55 7E 54 C4 17 2D 3F 29 46 53 42 44 C1 07 00 F8 A2 C1 0A 00 F8 ..,U~TÄ.-?)FSBDÄ..øçÄ..ø
000004440: A3 53 FB 0A 01 01 00 08 00 00 01 CF 00 03 C1 FF FF F8 A2 C1 68 01 F8 A3 £Sú.....Ï..ÄÿÿøçÄh.ø£
000004458: 53 FA 68 01 00 00 01 01 01 66 00 5C 03 D5 01 02 00 02 00 5C FF ED 03 3F Súh.....f.\.Ö.....\ÿi.?
000004470: 03 CA 00 33 00 40 00 B0 40 20 0F 42 1F 42 2F 42 03 0B 09 17 34 0C 0C 01 .Ë.3.è.°@ .B.B/B....4...
000004488: 4D 34 10 0D 01 4D 34 18 0B 01 4D 34 10 0F 10 01 4C 34 B8 01 F7 40 0E 00 M4...M4...M4...L4,÷@..
0000044A0: 27 27 0F 00 01 0C 00 10 0F 01 4D 00 3A B8 FF F0 B3 0C 01 4D 3A B8 FF E8 ''.....M.:;ÿô³..M.;ÿè
0000044B8: B3 0D 01 4D 3A B8 FF E8 B3 0B 01 4D 3A B8 FF E8 B4 0F 10 01 4C 3A B8 02 ³..M.;ÿè³..M.;ÿè´...L:;..
0000044D0: 08 40 14 11 17 C5 00 34 10 34 02 3B C0 34 01 00 34 01 34 34 2E 41 2A B8 .@...Ä.4.4.;Ä4..4.44.A*..
0000044E8: 01 1B 40 0F D0 22 01 22 22 04 1D EA 2E 4A 3D DC 0C 4D 04 00 2F 3F ED 3F ..è.B"....."'.è.J=Ü.M../?i?
000004500: ED 12 39 2F 5D ED 11 12 39 2F 5D 71 5E 5D ED 01 2F E1 2B 2B 2B 2B C4 2B í.9/]i..9/]q^]i./á++++Ä+
000004518: 5E 5D 39 2F 10 E1 2B 2B 2B 2B 32 39 31 30 5E 5D 25 14 06 06 23 22 26 26 ^]9/.á++++2910^]i%...#"&&
000004530: 35 35 06 06 23 22 2E 02 35 34 3E 02 33 33 35 34 2E 02 23 22 0E 02 23 22 55..#"..54>.3354..#"..#"..
000004548: 2E 02 1C 34 3E 04 33 32 1E 02 15 07 23 22 0E 02 15 14 16 33 32 36 37 03 ..54>.32....#".....3267.
000004560: 3F 10 1C 1B 1A 1F 0E 3B 91 51 47 73 51 2D 3E 74 A8 69 7C 16 31 4E 39 3D ?.....;QGsq->t`i|].1N9=
000004578: 61 49 31 0C 08 0C 09 04 07 1B 42 56 66 34 61 88 54 26 A4 8D 44 64 42 1F aI1.....BVf4a^T&•DdB.
000004590: 59 50 41 6F 3D 18 0C 0C 06 06 0C 0C 5A 3F 46 25 46 66 41 4C 70 4A 25 46 YPAo=.....Z?F%FfALpJ%F
0000045A8: 34 50 35 1B 1D 23 1D 08 10 19 0F 19 1D 1A 23 1D 13 2C 55 7E 54 C4 17 2D 4P5..#.....#..,U~TÄ.-
0000045C0: 3F 29 46 53 42 44 C1 FF FF F8 A2 C1 74 00 F8 A3 53 FB 74 01 01 00 72 00 ?)FSBDÄÿÿøçÄt.ø£Sút...r.
0000045D8: 00 02 01 01 F2 00 02 00 03 B7 01 D2 05 7A 00 15 00 16 00 1B 40 0D 0A ..V.ö......Ö.z.....@..
0000045F0: 04 15 7F 05 DF 05 02 05 80 10 16 48 00 3F DE 1A CD 5D 01 2F 1A CD 31 30 è.□ß...€..H.?b.Í].?f10
000004608: 01 3E 03 33 32 16 16 06 07 07 0E 03 23 22 26 26 36 37 07 01 1E 06 10 17 .>.32.....#"&67.....
000004620: 22 18 1C 23 0E 02 08 B5 08 0D 11 1B 14 16 1A 0B 03 07 8D 05 55 0B 0E 08 "...#...µ.....•.U...
000004638: 04 07 0D 11 09 E9 0A 0C 07 03 06 0A 10 0B B7 C1 08 00 F8 A2 C1 22 00 F8é......Ä..øçÄ".ø
000004650: A3 53 FB 22 01 01 00 20 00 5C 03 D5 01 04 FF FF 00 5C FF ED 03 3F 05 7A £Sú"... .\.Ö..ÿÿ.\ÿi.?.z
000004668: 02 26 01 02 00 00 07 01 F2 01 1A 00 00 C1 09 00 F8 A2 C1 E4 01 F8 A3 .&.....ö......Ä..øçÄä.ø£
000004680: 53 FA E4 01 00 00 01 01 01 F2 00 35 03 C4 01 50 00 03 00 35 FE 93 03 89 Súä.....ä.5.Ä.P...5p`%..
000004698: 03 C8 00 3E 00 4E 00 5F 00 D0 40 15 12 2D 47 10 0F 01 4D 47 08 0C 01 4D .È.>.N._.è@...-G...MG...M
0000046B0: 47 20 10 13 01 4C 47 33 3F B8 FF F8 B3 0C 01 4D 3F B8 FF F0 B3 0F 01 4D G ...LG3?.ÿô³..M?;ÿô³..M
0000046C8: 3F B8 FF E0 B6 10 13 01 4C 3F 07 4F B8 FF E0 40 0D 0F 12 01 4C 4F 1B 04 ?;ÿà¶...L?.O;ÿà@....LO..
0000046E0: 0F 30 03 33 07 07 B8 FF E0 B3 10 01 4D 1B B8 FF C0 40 0D 10 01 4D 2D 33 .0.3...ÿà³..M..ÿà@...M-3
0000046F8: 07 1B 1B 07 33 2D 04 00 B8 FF E0 40 34 0B 01 4D 00 58 10 0F 01 4D 58 203-...ÿà@4...M.X...MX
000004710: 10 12 01 4C 58 2A 25 40 0B 12 01 4C 25 0F 30 07 03 38 4A DA 0C 53 E8 15 ...LX*%@...L%.0..8JÚ.Sè..
000004728: 15 60 38 03 E9 3C 48 2A 20 42 D9 38 4A 5B D9 20 4F 00 3F ED 3F ED 12 39 `8.é<H* BÙ8J[Ü O.?.i?i.9
000004740: 3F ED 11 12 39 2F ED D6 ED 12 17 39 01 2F 2B 33 CD 2B 2B C4 2B 17 39 2F ?i..9/iöi..9./+3í++Ä+.9/
000004758: 2F 2F 2F 2B 2B 11 12 17 39 10 CD 2B 10 CD 2B 2B 2B 10 CD 10 CD 10 CD ///++...9.í+.í+++í+++í
000004770: 31 30 01 14 06 23 23 16 16 15 14 0E 02 23 22 26 27 06 06 15 14 16 17 17 10.....#"&'.....
000004788: 1E 03 15 14 0E 02 23 22 2E 02 35 34 3E 02 37 26 26 35 34 36 37 26 26 35#"..54>.7&5467&5
0000047A0: 34 3E 02 33 32 16 17 21 32 16 01 34 26 23 22 0E 02 15 14 16 33 32 3E 02 4>.32...!2..4&#".....32>..
0000047B8: 13 34 26 27 27 0E 03 15 14 16 33 32 3E 02 03 89 13 0F 81 23 1C 32 5D 80 .4&''.32>?..%...•.#.2]€
0000047D0: 4E 37 63 1B 12 17 41 36 EB 43 70 52 2E 34 6B A6 71 6D 99 62 2C 10 21 31 N7c...A6èCpR.4k|qm™b,;!1
0000047E8: 20 31 2F 32 25 1F 24 34 5C 81 4C 29 47 1F 01 10 11 11 FE FE 65 5D 30 47 1/2%. \$4(•L)G....bpe]0G
000004800: 2F 17 65 5B 31 48 2F 17 53 5F 51 E9 20 29 18 09 84 76 4B 65 3E 1B 03 74 /e[1ß/.S_Qé)...„vKe>...t
000004818: 23 1F 24 57 2F 4E 78 53 2C 1D 16 12 2E 1E 23 2E 02 0A 02 21 3D 59 3B 3E #\$. \$W/NxS,.....#.....!=Y;>


```

000004ED8: 00 B8 42 F8 A6 C1 02 00 F8 AA 6F D5 00 00 80 3F 00 00 80 3F F8 A4 65 D3 .,Bø!Á..ø*oÕ..e?..e?øneó
000004EF0: 61 02 41 02 F8 4C 6B C9 C1 0D 00 0B 00 05 00 09 00 0C 00 0A 00 0D 00 06 a.A.øLkÉÁ.....
000004F08: 00 01 00 08 00 09 00 03 00 0A 00 06 00 F8 AB C8 C1 0D 00 2A 2E 2B 30 30 .....ø«ÉÁ...*.+00
000004F20: 30 2C 30 2C 2B 15 30 00 F8 AF A8 85 C8 C1 03 00 00 00 00 F8 0B 63 D3 98 0,0,+.0.ø"....ÉÁ.....ø.cÓ~
000004F38: 04 41 02 F8 4C 6B C9 C1 01 00 07 00 F8 AB A8 C1 01 00 F8 31 44 C8 C1 10 .A.øLkÉÁ....ø«"Á..ø1DÉÁ.
000004F50: 00 4D 53 20 50 43 4C 58 4C 46 6F 6E 74 20 30 30 31 F8 A8 55 49 42 1B 25 .MS PCLXLFont 001ø"UIB.%
000004F68: 2D 31 32 33 34 35 58 40 50 4A 4C 20 45 4F 4A 0A 1B 25 2D 31 32 33 34 35 -12345X@PJL EOJ..%-12345
000004F80: 58 X

```

2.2.5.4.1.2 Análisis JPL/PCLXL

A continuación se muestra el análisis del fichero de spool generado al imprimir el documento en una impresora HP LaserJet 5500 que emplea lenguaje PJP/PCLXL.

Pero antes veamos algunos detalles de la estructura del documento:

- Empieza y acaba (0x00000000, 0x00004f78) con el comando UEL (<ESC>-12345X). Es decir utiliza lenguaje de control PJP
- Dentro del bloque PJP se definen propiedades del trabajo de impresión, como:
 - o el nombre del documento: Ejemplo.docx (Pos. 0x00000009, Pos. 0x000000d3)
 - o Aplicación desde la que se imprime: Microsoft Word (Pos. 0x00000330)
 - o usuario y dominio: miguel, WUAL (Pos. 0x00000278).
 - o Impresora por el que se lanza: HP LaserJet 5500 PCL 6 (Pos. 0x00000053)
 - o El servidor de impresión: DUKE3D. (Pos. 0x00000255)
 - o La fecha de impresión: 18/05/2016 13:15:29 (20160518131529). (Pos. 0x00000299)
 - o Si se utiliza página para separar el trabajo (SET SEPARATORPAGE=OFF). (Pos. 0x00000299)
 - o Etc.
 - o En la posición 0x00000640 aparece la instrucción @PJP ENTER LANGUAGE=PCLXL, que indica que se va a conmutar a lenguaje PCLXL para describir las páginas del documento, hasta que se vuelve a conmutar de nuevo a PJP, en la posición 0x00004f66 con el comando UEL (<ESC>-12345X).
- El bloque PCLXL está comprendido entre la posición: 0x00000659 (después de la instrucción que indica que se va a conmutar a lenguaje PJP: @PJP ENTER LANGUAGE=PCLXL) y la posición: 0x00004f65 (antes del siguiente comando UEL).
- Todo el contenido PCLXL esta definido entre los operadores BeginSession (Pos. 0x000006a9) y EndSession (Pos. 0x00004f65)
- Dentro de la sesión cada página se define, según el orden de impresión, entre los operadores BeginPage (Pos. 0x000006c6 y Pos. 0x00004a2f) y EndPage(Pos. 0x00004a23 y Pos. 0x00004f4c). Por lo que comprobamos que el documento consta de 2 páginas.
- Como ejemplo de análisis, si en el contenido PCLXL dentro de cada página se busca el atributo RGBColor, empleado para establecer el color que emplean los objetos de dibujo, podemos identificar qué páginas tienen contenido en color. Cuando el valor del atributo es distinto de negro (0, 0, 0) en hexadecimal 3 bytes (0x00, 0x00, 0x00) o blanco (255, 255, 255) en hexadecimal 3 bytes (0xff, 0xff, 0xff) quiere decir que el contenido de la página tiene color. En el ejemplo mostrado la primera página es en b/n y la segunda es en color.

Sin más se muestra el log con el análisis del fichero de spool generado desde el servicio Monitor de Impresión desarrollado en este TFG gracias a la librería log4net.


```

Posición 0x00004F66: ESC%-12345X
Posición 0x00004F6F: @PJL EOJ
Posición 0x00004F78: ESC%-12345X
Tiempo transcurrido en Milisegundos del analisis JPL: 24 ms
Cerramos el fichero analizado mediante lenguaje JPL
/*-----*/
                        FIN ANALISIS JPL
/*-----*/
OBJETO DE LA CLASE PCLXLSpool. Fichero en siguiente ruta:
D:\SpoolImpresora\Test\Prueba\EjemploPCLXL.SPL
Tamaño de fichero de Spool PCLXL: 20353
/*-----*/
                        INICIO ANALISIS PCLXL
/*-----*/
Fichero de Spool a analizar contenido PCLXL: D:\SpoolImpresora\Test\Prueba\EjemploPCLXL.SPL
Desde la posición: 0x00000659, hasta la posición: 0x00004F65
Posición: (0x0000065A): CABECERA: ' ) HP-PCL XL;2;1;Comment Copyright(c) 1999 Microsoft Corporation'
Posición: (0x0000069A): - Comienza analisis Bloque datos PCLXL.
  Pos. 0x0000069A    PCLXL Data Type    0xD1    uint16_xy
  Pos. 0x0000069B          Value    0x58025802
  Pos. 0x0000069F    PCLXL Attribute    0x89    UnitsPerMeasure
  Pos. 0x000006A1    PCLXL Data Type    0xC0    ubyte
  Pos. 0x000006A2          Value    0x00
  Pos. 0x000006A3    PCLXL Attribute    0x86    Measure
  Pos. 0x000006A5    PCLXL Data Type    0xC0    ubyte
  Pos. 0x000006A6          Value    0x03
  Pos. 0x000006A7    PCLXL Attribute    0x8F    ErrorReport
  Pos. 0x000006A9    PCLXL Operator ===>>>    0x41    BeginSession
-----
  Pos. 0x000006AA    PCLXL Data Type    0xC0    ubyte
  Pos. 0x000006AB          Value    0x00
  Pos. 0x000006AC    PCLXL Attribute    0x88    SourceType
  Pos. 0x000006AE    PCLXL Data Type    0xC0    ubyte
  Pos. 0x000006AF          Value    0x01
  Pos. 0x000006B0    PCLXL Attribute    0x82    DataOrg
  Pos. 0x000006B2    PCLXL Operator ===>>>    0x48    OpenDataSource
-----
MediaSize
  Pos. 0x000006B3    PCLXL Data Type    0xC0    ubyte
  Pos. 0x000006B4          Value    0x01
  Pos. 0x000006B5    PCLXL Attribute    0x26    MediaSource
  Pos. 0x000006B7    PCLXL Data Type    0xC0    ubyte
  Pos. 0x000006B8          Value    0x01
  Pos. 0x000006B9    PCLXL Attribute    0x35    DuplexPageMode
  Pos. 0x000006BB    PCLXL Data Type    0xC0    ubyte
  Pos. 0x000006BC          Value    0x00
  Pos. 0x000006BD    PCLXL Attribute    0x28    Orientation
  Pos. 0x000006BF    PCLXL Data Type    0xC8    ubyte_array
  Pos. 0x000006C2          Value    2 bytes
  Pos. 0x000006C4    PCLXL Attribute    0x25    MediaSize
  Pos. 0x000006C6    PCLXL Operator ===>>>    0x43    BeginPage
-----
  Pos. 0x000006C7    PCLXL Data Type    0xD3    sint16_xy
  Pos. 0x000006C8          Value    0x64006400
  Pos. 0x000006CC    PCLXL Attribute    0x2A    PageOrigin
  Pos. 0x000006CE    PCLXL Operator ===>>>    0x75    SetPageOrigin
-----
  Pos. 0x000006CF    PCLXL Data Type    0xD5    real32_xy
  Pos. 0x000006D0          Value    0x0000803F0000803F
  Pos. 0x000006D8    PCLXL Attribute    0x2B    PageScale
  Pos. 0x000006DA    PCLXL Operator ===>>>    0x77    SetPageScale
-----
  Pos. 0x000006DB    PCLXL Data Type    0xC0    ubyte
  Pos. 0x000006DC          Value    eRGB
  Pos. 0x000006DD    PCLXL Attribute    0x03    ColorSpace
  Pos. 0x000006DF    PCLXL Operator ===>>>    0x6A    SetColorSpace
-----
  Pos. 0x000006E0    PCLXL Data Type    0xC0    ubyte
  Pos. 0x000006E1          Value    0x00
  Pos. 0x000006E2    PCLXL Attribute    0x2D    TxMode
  Pos. 0x000006E4    PCLXL Operator ===>>>    0x78    SetPatternTxMode

```

Pos. 0x000006E5	PCLXL Data Type	0xC0	ubyte
Pos. 0x000006E6	Value	0x00	
Pos. 0x000006E7	PCLXL Attribute	0x2D	TxMode
Pos. 0x000006E9	PCLXL Operator ===>>>	0x7C	SetSourceTxMode
Pos. 0x000006EA	PCLXL Data Type	0xC0	ubyte
Pos. 0x000006EB	Value	0xFC	
Pos. 0x000006EC	PCLXL Attribute	0x2C	ROP3
Pos. 0x000006EE	PCLXL Operator ===>>>	0x7B	SetROP
Pos. 0x000006EF	PCLXL Operator ===>>>	0x85	NewPath
Pos. 0x000006F0	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x000006F4	Value	(0, 0, 0)	
Pos. 0x000006F7	PCLXL Attribute	0x0B	RGBColor
Pos. 0x000006F9	PCLXL Operator ===>>>	0x63	SetBrushSource
Pos. 0x000006FA	PCLXL Data Type	0xC0	ubyte
Pos. 0x000006FB	Value	0x00	
Pos. 0x000006FC	PCLXL Attribute	0xA9	FontFormat
Pos. 0x000006FE	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x00000702	Value	16 bytes	
Pos. 0x00000712	PCLXL Attribute	0xA8	FontName
Pos. 0x00000714	PCLXL Operator ===>>>	0x4F	BeginFontHeader
Pos. 0x00000715	PCLXL Data Type	0xC1	uint16
Pos. 0x00000716	Value	0x0800	
Pos. 0x00000718	PCLXL Attribute	0xA7	FontHeaderLength
Pos. 0x0000071A	PCLXL Operator ===>>>	0x50	ReadFontHeader
Pos. 0x0000071B	PCLXL Data Type	0xFB	dataLengthByte
Pos. 0x0000071C	PCLXL Embedded Len.	0x00000008	
Pos. 0x0000071D	Data	[8 bytes]	
Pos. 0x00000725	PCLXL Data Type	0xC1	uint16
Pos. 0x00000726	Value	0x0600	
Pos. 0x00000728	PCLXL Attribute	0xA7	FontHeaderLength
Pos. 0x0000072A	PCLXL Operator ===>>>	0x50	ReadFontHeader
Pos. 0x0000072B	PCLXL Data Type	0xFB	dataLengthByte
Pos. 0x0000072C	PCLXL Embedded Len.	0x00000006	
Pos. 0x0000072D	Data	[6 bytes]	
Pos. 0x00000733	PCLXL Data Type	0xC1	uint16
Pos. 0x00000734	Value	0x0C00	
Pos. 0x00000736	PCLXL Attribute	0xA7	FontHeaderLength
Pos. 0x00000738	PCLXL Operator ===>>>	0x50	ReadFontHeader
Pos. 0x00000739	PCLXL Data Type	0xFB	dataLengthByte
Pos. 0x0000073A	PCLXL Embedded Len.	0x0000000C	
Pos. 0x0000073B	Data	[12 bytes]	
Pos. 0x00000747	PCLXL Data Type	0xC1	uint16
Pos. 0x00000748	Value	0x6000	
Pos. 0x0000074A	PCLXL Attribute	0xA7	FontHeaderLength
Pos. 0x0000074C	PCLXL Operator ===>>>	0x50	ReadFontHeader
Pos. 0x0000074D	PCLXL Data Type	0xFB	dataLengthByte
Pos. 0x0000074E	PCLXL Embedded Len.	0x00000060	
Pos. 0x0000074F	Data	[96 bytes]	
Pos. 0x000007AF	PCLXL Data Type	0xC1	uint16
Pos. 0x000007B0	Value	0x1C05	
Pos. 0x000007B2	PCLXL Attribute	0xA7	FontHeaderLength
Pos. 0x000007B4	PCLXL Operator ===>>>	0x50	ReadFontHeader
Pos. 0x000007B5	PCLXL Data Type	0xFA	dataLengthInteger
Pos. 0x000007B6	PCLXL Embedded Len.	0x0000051C	
Pos. 0x000007BA	Data	[1308 bytes]	
Pos. 0x00000CD6	PCLXL Data Type	0xC1	uint16
Pos. 0x00000CD7	Value	0x1C09	
Pos. 0x00000CD9	PCLXL Attribute	0xA7	FontHeaderLength
Pos. 0x00000CDB	PCLXL Operator ===>>>	0x50	ReadFontHeader

Pos. 0x00000CDC	PCLXL Data Type	0xFA	dataLengthInteger
Pos. 0x00000CDD	PCLXL Embedded Len.	0x0000091C	
Pos. 0x00000CE1	Data	[2332 bytes]	
Pos. 0x000015FD	PCLXL Data Type	0xC1	uint16
Pos. 0x000015FE	Value	0x3800	
Pos. 0x00001600	PCLXL Attribute	0xA7	FontHeaderLength
Pos. 0x00001602	PCLXL Operator ===>>>	0x50	ReadFontHeader

Pos. 0x00001603	PCLXL Data Type	0xFB	dataLengthByte
Pos. 0x00001604	PCLXL Embedded Len.	0x00000038	
Pos. 0x00001605	Data	[56 bytes]	
Pos. 0x0000163D	PCLXL Data Type	0xC1	uint16
Pos. 0x0000163E	Value	0x2000	
Pos. 0x00001640	PCLXL Attribute	0xA7	FontHeaderLength
Pos. 0x00001642	PCLXL Operator ===>>>	0x50	ReadFontHeader

Pos. 0x00001643	PCLXL Data Type	0xFB	dataLengthByte
Pos. 0x00001644	PCLXL Embedded Len.	0x00000020	
Pos. 0x00001645	Data	[32 bytes]	
Pos. 0x00001665	PCLXL Data Type	0xC1	uint16
Pos. 0x00001666	Value	0x0020	
Pos. 0x00001668	PCLXL Attribute	0xA7	FontHeaderLength
Pos. 0x0000166A	PCLXL Operator ===>>>	0x50	ReadFontHeader

Pos. 0x0000166B	PCLXL Data Type	0xFA	dataLengthInteger
Pos. 0x0000166C	PCLXL Embedded Len.	0x00002000	
Pos. 0x00001670	Data	[8192 bytes]	
Pos. 0x00003670	PCLXL Data Type	0xC1	uint16
Pos. 0x00003671	Value	0x1404	
Pos. 0x00003673	PCLXL Attribute	0xA7	FontHeaderLength
Pos. 0x00003675	PCLXL Operator ===>>>	0x50	ReadFontHeader

Pos. 0x00003676	PCLXL Data Type	0xFA	dataLengthInteger
Pos. 0x00003677	PCLXL Embedded Len.	0x00000414	
Pos. 0x0000367B	Data	[1044 bytes]	
Pos. 0x00003A8F	PCLXL Data Type	0xC1	uint16
Pos. 0x00003A90	Value	0x0C00	
Pos. 0x00003A92	PCLXL Attribute	0xA7	FontHeaderLength
Pos. 0x00003A94	PCLXL Operator ===>>>	0x50	ReadFontHeader

Pos. 0x00003A95	PCLXL Data Type	0xFB	dataLengthByte
Pos. 0x00003A96	PCLXL Embedded Len.	0x0000000C	
Pos. 0x00003A97	Data	[12 bytes]	
Pos. 0x00003AA3	PCLXL Data Type	0xC1	uint16
Pos. 0x00003AA4	Value	0x0600	
Pos. 0x00003AA6	PCLXL Attribute	0xA7	FontHeaderLength
Pos. 0x00003AA8	PCLXL Operator ===>>>	0x50	ReadFontHeader

Pos. 0x00003AA9	PCLXL Data Type	0xFB	dataLengthByte
Pos. 0x00003AAA	PCLXL Embedded Len.	0x00000006	
Pos. 0x00003AAB	Data	[6 bytes]	
Pos. 0x00003AB1	PCLXL Operator ===>>>	0x51	EndFontHeader

Pos. 0x00003AB2	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x00003AB6	Value	16 bytes	
Pos. 0x00003AC6	PCLXL Attribute	0xA8	FontName
Pos. 0x00003AC8	PCLXL Operator ===>>>	0x52	BeginChar

Pos. 0x00003AC9	PCLXL Data Type	0xC1	uint16
Pos. 0x00003ACA	Value	0xFFFF	
Pos. 0x00003ACC	PCLXL Attribute	0xA2	CharCode
Pos. 0x00003ACE	PCLXL Data Type	0xC1	uint16
Pos. 0x00003ACF	Value	0x4C01	
Pos. 0x00003AD1	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x00003AD3	PCLXL Operator ===>>>	0x53	ReadChar

Pos. 0x00003AD4	PCLXL Data Type	0xFA	dataLengthInteger
Pos. 0x00003AD5	PCLXL Embedded Len.	0x0000014C	
Pos. 0x00003AD9	Data	[332 bytes]	
Pos. 0x00003C25	PCLXL Operator ===>>>	0x54	EndChar

Pos. 0x00003C26	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x00003C2A	Value	16 bytes	
Pos. 0x00003C3A	PCLXL Attribute	0xA8	FontName
Pos. 0x00003C3C	PCLXL Data Type	0xC5	real32
Pos. 0x00003C3D	Value	0x0000B842	
Pos. 0x00003C41	PCLXL Attribute	0xA6	CharSize
Pos. 0x00003C43	PCLXL Data Type	0xC1	uint16
Pos. 0x00003C44	Value	0x0200	
Pos. 0x00003C46	PCLXL Attribute	0xAA	SymbolSet
Pos. 0x00003C48	PCLXL Operator ==>>>	0x6F	SetFont

Pos. 0x00003C49	PCLXL Data Type	0xD5	real32_xy
Pos. 0x00003C4A	Value	0x0000803F0000803F	
Pos. 0x00003C52	PCLXL Attribute	0xA4	CharScale
Pos. 0x00003C54	PCLXL Operator ==>>>	0x65	SetCharScale

Pos. 0x00003C55	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x00003C59	Value	16 bytes	
Pos. 0x00003C69	PCLXL Attribute	0xA8	FontName
Pos. 0x00003C6B	PCLXL Operator ==>>>	0x52	BeginChar

Pos. 0x00003C6C	PCLXL Data Type	0xC1	uint16
Pos. 0x00003C6D	Value	0x0100	
Pos. 0x00003C6F	PCLXL Attribute	0xA2	CharCode
Pos. 0x00003C71	PCLXL Data Type	0xC1	uint16
Pos. 0x00003C72	Value	0xFC00	
Pos. 0x00003C74	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x00003C76	PCLXL Operator ==>>>	0x53	ReadChar

Pos. 0x00003C77	PCLXL Data Type	0xFB	dataLengthByte
Pos. 0x00003C78	PCLXL Embedded Len.	0x000000FC	
Pos. 0x00003C79	Data	[252 bytes]	
Pos. 0x00003D75	PCLXL Data Type	0xC1	uint16
Pos. 0x00003D76	Value	0x0200	
Pos. 0x00003D78	PCLXL Attribute	0xA2	CharCode
Pos. 0x00003D7A	PCLXL Data Type	0xC1	uint16
Pos. 0x00003D7B	Value	0xFC00	
Pos. 0x00003D7D	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x00003D7F	PCLXL Operator ==>>>	0x53	ReadChar

Pos. 0x00003D80	PCLXL Data Type	0xFB	dataLengthByte
Pos. 0x00003D81	PCLXL Embedded Len.	0x000000FC	
Pos. 0x00003D82	Data	[252 bytes]	
Pos. 0x00003E7E	PCLXL Data Type	0xC1	uint16
Pos. 0x00003E7F	Value	0x0300	
Pos. 0x00003E81	PCLXL Attribute	0xA2	CharCode
Pos. 0x00003E83	PCLXL Data Type	0xC1	uint16
Pos. 0x00003E84	Value	0xFC00	
Pos. 0x00003E86	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x00003E88	PCLXL Operator ==>>>	0x53	ReadChar

Pos. 0x00003E89	PCLXL Data Type	0xFB	dataLengthByte
Pos. 0x00003E8A	PCLXL Embedded Len.	0x000000FC	
Pos. 0x00003E8B	Data	[252 bytes]	
Pos. 0x00003F87	PCLXL Data Type	0xC1	uint16
Pos. 0x00003F88	Value	0x0400	
Pos. 0x00003F8A	PCLXL Attribute	0xA2	CharCode
Pos. 0x00003F8C	PCLXL Data Type	0xC1	uint16
Pos. 0x00003F8D	Value	0xCC01	
Pos. 0x00003F8F	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x00003F91	PCLXL Operator ==>>>	0x53	ReadChar

Pos. 0x00003F92	PCLXL Data Type	0xFA	dataLengthInteger
Pos. 0x00003F93	PCLXL Embedded Len.	0x000001CC	
Pos. 0x00003F97	Data	[460 bytes]	
Pos. 0x00004163	PCLXL Data Type	0xC1	uint16
Pos. 0x00004164	Value	0x0500	
Pos. 0x00004166	PCLXL Attribute	0xA2	CharCode
Pos. 0x00004168	PCLXL Data Type	0xC1	uint16
Pos. 0x00004169	Value	0x4C01	
Pos. 0x0000416B	PCLXL Attribute	0xA3	CharDataSize

Pos. 0x0000416D	PCLXL Operator ==>>>	0x53	ReadChar

Pos. 0x0000416E	PCLXL Data Type	0xFA	dataLengthInteger
Pos. 0x0000416F	PCLXL Embedded Len.	0x0000014C	
Pos. 0x00004173	Data	[332 bytes]	
Pos. 0x000042BF	PCLXL Data Type	0xC1	uint16
Pos. 0x000042C0	Value	0x0600	
Pos. 0x000042C2	PCLXL Attribute	0xA2	CharCode
Pos. 0x000042C4	PCLXL Data Type	0xC1	uint16
Pos. 0x000042C5	Value	0x6801	
Pos. 0x000042C7	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x000042C9	PCLXL Operator ==>>>	0x53	ReadChar

Pos. 0x000042CA	PCLXL Data Type	0xFA	dataLengthInteger
Pos. 0x000042CB	PCLXL Embedded Len.	0x00000168	
Pos. 0x000042CF	Data	[360 bytes]	
Pos. 0x00004437	PCLXL Data Type	0xC1	uint16
Pos. 0x00004438	Value	0x0700	
Pos. 0x0000443A	PCLXL Attribute	0xA2	CharCode
Pos. 0x0000443C	PCLXL Data Type	0xC1	uint16
Pos. 0x0000443D	Value	0x0A00	
Pos. 0x0000443F	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x00004441	PCLXL Operator ==>>>	0x53	ReadChar

Pos. 0x00004442	PCLXL Data Type	0xFB	dataLengthByte
Pos. 0x00004443	PCLXL Embedded Len.	0x0000000A	
Pos. 0x00004444	Data	[10 bytes]	
Pos. 0x0000444E	PCLXL Data Type	0xC1	uint16
Pos. 0x0000444F	Value	0xFFFF	
Pos. 0x00004451	PCLXL Attribute	0xA2	CharCode
Pos. 0x00004453	PCLXL Data Type	0xC1	uint16
Pos. 0x00004454	Value	0x6801	
Pos. 0x00004456	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x00004458	PCLXL Operator ==>>>	0x53	ReadChar

Pos. 0x00004459	PCLXL Data Type	0xFA	dataLengthInteger
Pos. 0x0000445A	PCLXL Embedded Len.	0x00000168	
Pos. 0x0000445E	Data	[360 bytes]	
Pos. 0x000045C6	PCLXL Data Type	0xC1	uint16
Pos. 0x000045C7	Value	0xFFFF	
Pos. 0x000045C9	PCLXL Attribute	0xA2	CharCode
Pos. 0x000045CB	PCLXL Data Type	0xC1	uint16
Pos. 0x000045CC	Value	0x7400	
Pos. 0x000045CE	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x000045D0	PCLXL Operator ==>>>	0x53	ReadChar

Pos. 0x000045D1	PCLXL Data Type	0xFB	dataLengthByte
Pos. 0x000045D2	PCLXL Embedded Len.	0x00000074	
Pos. 0x000045D3	Data	[116 bytes]	
Pos. 0x00004647	PCLXL Data Type	0xC1	uint16
Pos. 0x00004648	Value	0x0800	
Pos. 0x0000464A	PCLXL Attribute	0xA2	CharCode
Pos. 0x0000464C	PCLXL Data Type	0xC1	uint16
Pos. 0x0000464D	Value	0x2200	
Pos. 0x0000464F	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x00004651	PCLXL Operator ==>>>	0x53	ReadChar

Pos. 0x00004652	PCLXL Data Type	0xFB	dataLengthByte
Pos. 0x00004653	PCLXL Embedded Len.	0x00000022	
Pos. 0x00004654	Data	[34 bytes]	
Pos. 0x00004676	PCLXL Data Type	0xC1	uint16
Pos. 0x00004677	Value	0x0900	
Pos. 0x00004679	PCLXL Attribute	0xA2	CharCode
Pos. 0x0000467B	PCLXL Data Type	0xC1	uint16
Pos. 0x0000467C	Value	0xE401	
Pos. 0x0000467E	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x00004680	PCLXL Operator ==>>>	0x53	ReadChar

Pos. 0x00004681	PCLXL Data Type	0xFA	dataLengthInteger
Pos. 0x00004682	PCLXL Embedded Len.	0x000001E4	
Pos. 0x00004686	Data	[484 bytes]	

Pos. 0x0000486A	PCLXL Data Type	0xC1	uint16
Pos. 0x0000486B	Value	0x0A00	
Pos. 0x0000486D	PCLXL Attribute	0xA2	CharCode
Pos. 0x0000486F	PCLXL Data Type	0xC1	uint16
Pos. 0x00004870	Value	0x2C01	
Pos. 0x00004872	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x00004874	PCLXL Operator ===>>>	0x53	ReadChar

Pos. 0x00004875	PCLXL Data Type	0xFA	dataLengthInteger
Pos. 0x00004876	PCLXL Embedded Len.	0x0000012C	
Pos. 0x0000487A	Data	[300 bytes]	
Pos. 0x000049A6	PCLXL Operator ===>>>	0x54	EndChar

Pos. 0x000049A7	PCLXL Data Type	0xD3	sint16_xy
Pos. 0x000049A8	Value	0x61024102	
Pos. 0x000049AC	PCLXL Attribute	0x4C	Point
Pos. 0x000049AE	PCLXL Operator ===>>>	0x6B	SetCursor

Pos. 0x000049AF	PCLXL Data Type	0xC9	uint16_array
Pos. 0x000049B3	Value	28 bytes	
Pos. 0x000049CF	PCLXL Attribute	0xAB	TextData
Pos. 0x000049D1	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x000049D5	Value	14 bytes	
Pos. 0x000049E3	PCLXL Attribute	0xAF	XSpacingData
Pos. 0x000049E5	PCLXL Operator ===>>>	0xA8	Text

Pos. 0x000049E6	PCLXL Operator ===>>>	0x85	NewPath

Pos. 0x000049E7	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x000049EB	Value	(0, 0, 0)	
Pos. 0x000049EE	PCLXL Attribute	0x0B	RGBColor
Pos. 0x000049F0	PCLXL Operator ===>>>	0x63	SetBrushSource

Pos. 0x000049F1	PCLXL Data Type	0xD3	sint16_xy
Pos. 0x000049F2	Value	0x94044102	
Pos. 0x000049F6	PCLXL Attribute	0x4C	Point
Pos. 0x000049F8	PCLXL Operator ===>>>	0x6B	SetCursor

Pos. 0x000049F9	PCLXL Data Type	0xC9	uint16_array
Pos. 0x000049FD	Value	2 bytes	
Pos. 0x000049FF	PCLXL Attribute	0xAB	TextData
Pos. 0x00004A01	PCLXL Operator ===>>>	0xA8	Text

Pos. 0x00004A02	PCLXL Operator ===>>>	0x85	NewPath

Pos. 0x00004A03	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x00004A07	Value	(0, 0, 0)	
Pos. 0x00004A0A	PCLXL Attribute	0x0B	RGBColor
Pos. 0x00004A0C	PCLXL Operator ===>>>	0x63	SetBrushSource

Pos. 0x00004A0D	PCLXL Data Type	0xD3	sint16_xy
Pos. 0x00004A0E	Value	0x44094102	
Pos. 0x00004A12	PCLXL Attribute	0x4C	Point
Pos. 0x00004A14	PCLXL Operator ===>>>	0x6B	SetCursor

Pos. 0x00004A15	PCLXL Data Type	0xC9	uint16_array
Pos. 0x00004A19	Value	2 bytes	
Pos. 0x00004A1B	PCLXL Attribute	0xAB	TextData
Pos. 0x00004A1D	PCLXL Operator ===>>>	0xA8	Text

*****NUMERO DE PAGINAS: 1			
*****NUMERO DE PAGINAS COLOR: 0			

Pos. 0x00004A1E	PCLXL Data Type	0xC1	uint16
Pos. 0x00004A1F	Value	0x0100	
Pos. 0x00004A21	PCLXL Attribute	0x31	PageCopies
Pos. 0x00004A23	PCLXL Operator ===>>>	0x44	EndPage

MediaSize			
Pos. 0x00004A24	PCLXL Data Type	0xC0	ubyte
Pos. 0x00004A25	Value	0x00	

Pos. 0x00004A26	PCLXL Attribute	0x28	Orientation
Pos. 0x00004A28	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x00004A2B	Value	2 bytes	
Pos. 0x00004A2D	PCLXL Attribute	0x25	MediaSize
Pos. 0x00004A2F	PCLXL Operator ===>>>	0x43	BeginPage

Pos. 0x00004A30	PCLXL Data Type	0xD3	sint16_xy
Pos. 0x00004A31	Value	0x64006400	
Pos. 0x00004A35	PCLXL Attribute	0x2A	PageOrigin
Pos. 0x00004A37	PCLXL Operator ===>>>	0x75	SetPageOrigin

Pos. 0x00004A38	PCLXL Data Type	0xD5	real32_xy
Pos. 0x00004A39	Value	0x0000803F0000803F	
Pos. 0x00004A41	PCLXL Attribute	0x2B	PageScale
Pos. 0x00004A43	PCLXL Operator ===>>>	0x77	SetPageScale

Pos. 0x00004A44	PCLXL Data Type	0xC0	ubyte
Pos. 0x00004A45	Value	eRGB	
Pos. 0x00004A46	PCLXL Attribute	0x03	ColorSpace
Pos. 0x00004A48	PCLXL Operator ===>>>	0x6A	SetColorSpace

Pos. 0x00004A49	PCLXL Data Type	0xC0	ubyte
Pos. 0x00004A4A	Value	0xCC	
Pos. 0x00004A4B	PCLXL Attribute	0x2C	ROP3
Pos. 0x00004A4D	PCLXL Operator ===>>>	0x7B	SetROP

Pos. 0x00004A4E	PCLXL Data Type	0xC0	ubyte
Pos. 0x00004A4F	Value	0xFC	
Pos. 0x00004A50	PCLXL Attribute	0x2C	ROP3
Pos. 0x00004A52	PCLXL Operator ===>>>	0x7B	SetROP

Pos. 0x00004A53	PCLXL Operator ===>>>	0x85	NewPath

Pos. 0x00004A54	PCLXL Data Type	0xC0	ubyte
Pos. 0x00004A55	Value	0x00	
Pos. 0x00004A56	PCLXL Attribute	0x2D	TxMode
Pos. 0x00004A58	PCLXL Operator ===>>>	0x78	SetPatternTxMode

Pos. 0x00004A59	PCLXL Data Type	0xC0	ubyte
Pos. 0x00004A5A	Value	0x00	
Pos. 0x00004A5B	PCLXL Attribute	0x2D	TxMode
Pos. 0x00004A5D	PCLXL Operator ===>>>	0x7C	SetSourceTxMode

Pos. 0x00004A5E	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x00004A62	Value	(0, 176, 80)	
Pos. 0x00004A65	PCLXL Attribute	0x0B	RGBColor
Pos. 0x00004A67	PCLXL Operator ===>>>	0x63	SetBrushSource

Pos. 0x00004A68	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x00004A6C	Value	16 bytes	
Pos. 0x00004A7C	PCLXL Attribute	0xA8	FontName
Pos. 0x00004A7E	PCLXL Operator ===>>>	0x52	BeginChar

Pos. 0x00004A7F	PCLXL Data Type	0xC1	uint16
Pos. 0x00004A80	Value	0x0B00	
Pos. 0x00004A82	PCLXL Attribute	0xA2	CharCode
Pos. 0x00004A84	PCLXL Data Type	0xC1	uint16
Pos. 0x00004A85	Value	0xAE01	
Pos. 0x00004A87	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x00004A89	PCLXL Operator ===>>>	0x53	ReadChar

Pos. 0x00004A8A	PCLXL Data Type	0xFA	dataLengthInteger
Pos. 0x00004A8B	PCLXL Embedded Len.	0x000001AE	
Pos. 0x00004A8F	Data	[430 bytes]	
Pos. 0x00004C3D	PCLXL Data Type	0xC1	uint16
Pos. 0x00004C3E	Value	0x0C00	
Pos. 0x00004C40	PCLXL Attribute	0xA2	CharCode
Pos. 0x00004C42	PCLXL Data Type	0xC1	uint16
Pos. 0x00004C43	Value	0x2A01	
Pos. 0x00004C45	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x00004C47	PCLXL Operator ===>>>	0x53	ReadChar

Pos. 0x00004C48	PCLXL Data Type	0xFA	dataLengthInteger
Pos. 0x00004C49	PCLXL Embedded Len.	0x0000012A	
Pos. 0x00004C4D	Data	[298 bytes]	
Pos. 0x00004D77	PCLXL Data Type	0xC1	uint16
Pos. 0x00004D78	Value	0x0D00	
Pos. 0x00004D7A	PCLXL Attribute	0xA2	CharCode
Pos. 0x00004D7C	PCLXL Data Type	0xC1	uint16
Pos. 0x00004D7D	Value	0x3801	
Pos. 0x00004D7F	PCLXL Attribute	0xA3	CharDataSize
Pos. 0x00004D81	PCLXL Operator ===>>>	0x53	ReadChar
Pos. 0x00004D82	PCLXL Data Type	0xFA	dataLengthInteger
Pos. 0x00004D83	PCLXL Embedded Len.	0x00000138	
Pos. 0x00004D87	Data	[312 bytes]	
Pos. 0x00004EBF	PCLXL Operator ===>>>	0x54	EndChar
Pos. 0x00004EC0	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x00004EC4	Value	16 bytes	
Pos. 0x00004ED4	PCLXL Attribute	0xA8	FontName
Pos. 0x00004ED6	PCLXL Data Type	0xC5	real32
Pos. 0x00004ED7	Value	0x0000B842	
Pos. 0x00004EDB	PCLXL Attribute	0xA6	CharSize
Pos. 0x00004EDD	PCLXL Data Type	0xC1	uint16
Pos. 0x00004EDE	Value	0x0200	
Pos. 0x00004EE0	PCLXL Attribute	0xAA	SymbolSet
Pos. 0x00004EE2	PCLXL Operator ===>>>	0x6F	SetFont
Pos. 0x00004EE3	PCLXL Data Type	0xD5	real32_xy
Pos. 0x00004EE4	Value	0x0000803F0000803F	
Pos. 0x00004EEC	PCLXL Attribute	0xA4	CharScale
Pos. 0x00004EEE	PCLXL Operator ===>>>	0x65	SetCharScale
Pos. 0x00004EEF	PCLXL Data Type	0xD3	sint16_xy
Pos. 0x00004EF0	Value	0x61024102	
Pos. 0x00004EF4	PCLXL Attribute	0x4C	Point
Pos. 0x00004EF6	PCLXL Operator ===>>>	0x6B	SetCursor
Pos. 0x00004EF7	PCLXL Data Type	0xC9	uint16_array
Pos. 0x00004EFB	Value	26 bytes	
Pos. 0x00004F15	PCLXL Attribute	0xAB	TextData
Pos. 0x00004F17	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x00004F1B	Value	13 bytes	
Pos. 0x00004F28	PCLXL Attribute	0xAF	XSpacingData
Pos. 0x00004F2A	PCLXL Operator ===>>>	0xA8	Text
Pos. 0x00004F2B	PCLXL Operator ===>>>	0x85	NewPath
Pos. 0x00004F2C	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x00004F30	Value	(0, 0, 0)	
Pos. 0x00004F33	PCLXL Attribute	0x0B	RGBColor
Pos. 0x00004F35	PCLXL Operator ===>>>	0x63	SetBrushSource
Pos. 0x00004F36	PCLXL Data Type	0xD3	sint16_xy
Pos. 0x00004F37	Value	0x98044102	
Pos. 0x00004F3B	PCLXL Attribute	0x4C	Point
Pos. 0x00004F3D	PCLXL Operator ===>>>	0x6B	SetCursor
Pos. 0x00004F3E	PCLXL Data Type	0xC9	uint16_array
Pos. 0x00004F42	Value	2 bytes	
Pos. 0x00004F44	PCLXL Attribute	0xAB	TextData
Pos. 0x00004F46	PCLXL Operator ===>>>	0xA8	Text
*****NUMERO DE PAGINAS: 2			
*****NUMERO DE PAGINAS COLOR: 1			
Pos. 0x00004F47	PCLXL Data Type	0xC1	uint16
Pos. 0x00004F48	Value	0x0100	
Pos. 0x00004F4A	PCLXL Attribute	0x31	PageCopies
Pos. 0x00004F4C	PCLXL Operator ===>>>	0x44	EndPage

Pos. 0x00004F4D	PCLXL Data Type	0xC8	ubyte_array
Pos. 0x00004F51	Value	16 bytes	
Pos. 0x00004F61	PCLXL Attribute	0xA8	FontName
Pos. 0x00004F63	PCLXL Operator ==>>>	0x55	RemoveFont

Pos. 0x00004F64	PCLXL Operator ==>>>	0x49	CloseDataSource

Pos. 0x00004F65	PCLXL Operator ==>>>	0x42	EndSession

Posición: (0x00004F65): - Finaliza analisis Bloque datos PCLXL.			

Resumen Tags con el número de ocurrencias.			

BeginChar: 3			
BeginFontHeader: 1			
BeginPage: 2			
BeginSession: 1			
CloseDataSource: 1			
EndChar: 3			
EndFontHeader: 1			
EndPage: 2			
EndSession: 1			
NewPath: 5			
OpenDataSource: 1			
ReadChar: 16			
ReadFontHeader: 12			
RemoveFont: 1			
SetBrushSource: 5			
SetCharScale: 2			
SetColorSpace: 2			
SetCursor: 5			
SetFont: 2			
SetPageOrigin: 2			
SetPageScale: 2			
SetPatternTxMode: 2			
SetROP: 3			
SetSourceTxMode: 2			
Text: 5			

Total: 82			

/*-----*/			
FIN ANALISIS PCLXL			
/*-----*/			

2.2.5.4.2 EMFSPOOL/EMF

2.2.5.4.2.1 Contenido fichero de Spool:

Cada posición de memoria representa el contenido de un byte en hexadecimal. La columna de la derecha la representa el contenido hexadecimal en formato ASCII si es un carácter.

Offset hex.	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	
000000000:	00	00	01	00	4C	00	00	00	10	00	00	00	00	00	00	00	4D	00	69	00	63	00	72	00L.....M.i.c.r.
000000018:	6F	00	73	00	6F	00	66	00	74	00	20	00	57	00	6F	00	72	00	64	00	20	00	2D	00	o.s.o.f.t. .W.o.r.d. .-.
000000030:	20	00	45	00	6A	00	65	00	6D	00	70	00	6C	00	6F	00	2E	00	64	00	6F	00	63	00	.E.j.e.m.p.l.o...d.o.c.
000000048:	78	00	00	00	0C	00	00	00	18	09	00	00	01	00	00	00	84	00	00	00	61	02	00	00	x.....a...
000000060:	E9	01	00	00	79	09	00	00	59	02	00	00	00	00	00	00	00	00	00	84	4E	00	00	00	é...y...Y.....N..
000000078:	80	70	00	00	20	45	4D	46	00	00	01	00	18	09	00	00	26	00	00	00	05	00	00	00	€p... EMF.....&.....
000000090:	0C	00	00	00	6C	00	00	00	00	00	00	00	96	12	00	00	98	1A	00	00	C9	00	00	00l.....-....~....É...
0000000A8:	20	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	DC	12	03	00	DC	65	04	00Ü...Üe..
0000000C0:	50	00	72	00	69	00	6E	00	74	00	20	00	74	00	65	00	73	00	74	00	00	00	00	00	P.r.i.n.t. .t.e.s.t.....
0000000D8:	25	00	00	00	0C	00	00	00	07	00	00	80	25	00	00	00	0C	00	00	00	00	00	00	80	%.....€%.....€
0000000F0:	52	00	00	00	70	01	00	00	01	00	00	00	AD	FF	FF	FF	00	00	00	00	00	00	00	00	R...p.....-ÿÿÿ.....
000000108:	00	00	00	00	90	01	00	00	00	00	00	00	04	40	00	10	54	00	69	00	6D	00	65	00•.....@...T.i.m.e.
000000120:	73	00	20	00	4E	00	65	00	77	00	20	00	52	00	6F	00	6D	00	61	00	6E	00	00	00	s. .N.e.w. .R.o.m.a.n...
000000138:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000150:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000168:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000180:	00	00	00	00	00	00	00	00	00	00	00	00	2C	8E	2C	00	C4	8F	2C	00	1C	00	55	01,Ž,.Ä•,...U.
000000198:	60	8C	2C	00	07	61	63	31	C4	8F	2C	00	20	00	00	00	B8	8C	2C	00	FF	FF	FF	FF	`E,...aclÄ•,...CE,.ÿÿÿÿ
0000001B0:	A8	8F	2C	00	40	8D	2C	00	52	1B	6C	31	B8	8C	2C	00	C4	8F	2C	00	20	00	00	00	"•,...@•,.R.l1,CE,.Ä•,...
0000001C8:	FF	FF	FF	FF	5C	2B	54	01	C9	1B	6C	31	01	00	00	00	00	00	00	00	58	02	00	00	ÿÿÿÿ\+T.É.l1.....X...
0000001E0:	00	00	00	00	47	1E	90	01	00	00	00	00	00	00	00	00	00	00	00	00	FF	2A	00	E0G.~.....ÿ*.à
0000001F8:	41	78	00	C0	09	00	00	00	00	00	00	00	FF	01	00	00	00	00	00	00	54	00	69	00	Ax.Ä.....ÿ.....T.i.
000000210:	6D	00	65	00	73	00	20	00	00	00	65	00	77	00	20	00	52	00	6F	00	6D	00	61	00	m.e.s.e.w. .R.o.m.a.
000000228:	6E	00	00	00	E8	8C	2C	00	79	2A	66	31	08	00	00	00	FF	FF	FF	DF	24	8D	2C	00	n...èE,.y*f1....ÿÿÿB\$•,...
000000240:	2A	73	64	31	38	8D	2C	00	A0	01	57	32	50	8D	2C	00	5C	2B	54	01	80	2A	54	01	*sd18•,...W2P•,...\+T.€*T.
000000258:	64	76	00	08	00	00	00	00	25	00	00	00	0C	00	00	00	01	00	00	00	16	00	00	00	dv.....%.....
000000270:	0C	00	00	00	18	00	00	00	1B	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00
000000288:	0D	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00	62	00	00	00	0C	00	00	00b.....
0000002A0:	01	00	00	00	64	00	00	00	0C	00	00	00	14	00	00	80	6A	00	00	00	10	00	00	00d.....ej.....
0000002B8:	3C	0F	00	00	00	00	00	00	0C	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00	<.....
0000002D0:	6A	00	00	00	10	00	00	00	14	00	00	00	00	00	00	00	25	00	00	00	0C	00	00	00	j.....%.....
0000002E8:	0E	00	00	80	16	00	00	00	0C	00	00	00	18	00	00	00	0C	00	00	00	10	00	00	00	...€.....
000000300:	00	00	00	00	00	00	00	00	12	00	00	00	0C	00	00	00	01	00	00	00	52	00	00	00R...
000000318:	70	01	00	00	02	00	00	00	A4	FF	FF	FF	00	00	00	00	00	00	00	00	00	00	00	00	p.....#ÿÿÿ.....
000000330:	90	01	00	00	00	00	00	04	40	00	22	43	00	61	00	6C	00	69	00	62	00	72	00	00	•.....@."C.a.l.i.b.r.
000000348:	69	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	i.....
000000360:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000378:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000390:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000003A8:	00	00	00	00	00	00	00	00	D8	E0	DD	0D	70	E2	DD	0D	70	8B	55	01	0C	DF	DD	0DØäÿ.päÿ.p.U..ßÿ.
0000003C0:	07	61	63	31	70	E2	DD	0D	20	00	00	00	64	DF	DD	0D	FF	FF	FF	FF	54	E2	DD	0D	..aclpäÿ. ...dßÿ.ÿÿÿTäÿ.
0000003D8:	EC	DF	DD	0D	52	1B	6C	31	64	DF	DD	0D	70	E2	DD	0D	20	00	00	00	FF	FF	FF	FF	ißÿ.R.l1dßÿ.päÿ. ...ÿÿÿÿ
0000003F0:	6C	2C	54	01	C9	1B	6C	31	01	00	00	00	00	00	00	00	58	02	00	00	25	00	00	00	l,T.É.l1.....X...%...
000000408:	37	2E	90	01	00	00	00	00	00	00	00	00	00	00	00	00	FF	02	00	E0	FF	AC	00	40	7.~.....ÿ..àÿ~.@
000000420:	01	00	00	00	00	00	00	00	9F	01	00	00	00	00	00	00	43	00	61	00	6C	00	69	00ÿ.....C.a.l.i.
000000438:	62	00	72	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	b.r.....
000000450:	94	DF	DD	0D	79	2A	66	31	08	00	00	00	FF	FF	FF	DF	0D	DF	DD	0D	2A	73	64	31	"ßÿ.y*f1....ÿÿÿßDßÿ.*sd1
000000468:	E4	DF	DD	0D	A0	01	57	32	FC	DF	DD	0D	6C	2C	54	01	90	2B	54	01	64	76	00	08	äßÿ. .W2üßÿ.l,T.~+T.dv..
000000480:	00	00	00	00	25	00	00	00	0C	00	00	00	02	00	00	00	25	00	00	00	0C	00	00	00%.....%.....
000000498:	02	00	00	00	52	00	00	00	70	01	00	00	03	00	00	00	A4	FF	FF	FF	00	00	00	00R...p.....#ÿÿÿ....
0000004B0:	00	00	00	00	00	00	00	00	90	01	00	00	00	00	00	EE	09	40	00	22	43	00	61	00•.....f.@."C.a.
0000004C8:	6C	00	69	00	62	00	72	00	69	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	l.i.b.r.i.....
0000004E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000004F8:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	80	E3	DD	0Deäÿ.
000000510:	A5	17	16	76	F5	0E	01	24	10	09	98	07	00	00	00	00	01	00	00	00	F5	0E	01	24	¥..vø..\$.~.....ø..\$
000000528:	F5	0E	01	24	A0	E3	DD	0D	46	18	16	76	71	58	F6	38	00	00	00	00	80	0B	55	00	ø..\$ äÿ.F.vqÿø8....€U.
000000540:	00	60	5F	00	60	13	A1	0A	AC	E3	DD	0D	CB	56	F6	38	B8	45	44	00	60	00	00	00	..`_.`.j.-äÿ.Èvø8,ED.`...
000000558:	E8	E3	DD	0D	8F	56	F6	38	60	00	00	00	00	00	00	00	94	56	F6	38	74	61	F7	09	eäÿ.~vø8`....."vø8ta÷.
000000570:	60	00	00	00	B8	45	44	00	00	60	5F	00	52	E4	DD	0D	C4	45	44	00	00	00	00	00	`....ED.`_.Räÿ.ÆED...ÿ.
000000588:	B8	FE	DD	0D	D4	3A	F0	39	00	00	00	00	94	56	F6	38	54	56	F6	38	60	00	00	00	..ßÿ.ô:ø9...."vø8TVø8`...

0000005A0: 00 00 00 00 00 00 00 00 30 56 F6 38 60 00 00 00 80 00 00 00 3B 11 21 B30Vø8`...€...;!³
0000005B8: BC 13 A1 0A 00 60 5F 00 80 00 00 00 00 DD 0D 4B 7C F7 38 AC E5 F6 39 ¼.¡..`_€......Ý.K|÷8-äø9
0000005D0: 8C E4 DD 0D 1F 7C 6A 39 00 60 5F 00 AC E5 F6 39 01 00 00 00 2E 7C 6A 39 ðäÝ..|j9.`_.-äø9....|j9
0000005E8: F8 E4 DD 0D 01 00 00 00 43 00 61 00 6C 00 69 00 62 00 72 00 69 00 00 00 øäÝ.....C.a.l.i.b.r.i...
000000600: 00 00 00 00 64 76 00 08 00 00 00 00 25 00 00 00 0C 00 00 00 03 00 00 00dv.....%.....
000000618: 52 00 00 00 70 01 00 00 04 00 00 00 A4 FF FF FF 00 00 00 00 00 00 00 00 R...p.....nýÿÿ.....
000000630: 00 00 00 00 90 01 00 00 00 00 00 00 04 40 00 22 43 00 61 00 6C 00 69 00•.....@."C.a.l.i.
000000648: 62 00 72 00 69 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 b.r.i.....
000000660: 00
000000678: 00 00 00 00 00 00 00 00 00 00 F6 38 B8 FE DD 0D 45 E1 EC 76 D3 DD 5B 37ø8.pÝ.EáivÓÝ[7
000000690: FE FF FF FF CF 2B F0 76 6A 2A F0 76 1F 06 12 00 00 00 00 00 F8 6B C0 07 þÿÿÿÿ+ðvj*ðv.....økÀ.
0000006A8: B8 45 44 00 00 60 5F 00 52 E4 DD 0D C4 45 44 00 00 00 DD 0D B8 FE DD 0D ,ED..`_..RäÝ.ÆED...Ý..þÝ.
0000006C0: D4 3A F0 39 00 00 00 94 56 F6 38 54 56 F6 38 60 00 00 00 00 00 00 00 00 Ô:ø9...."Vø8TVø8`.....
0000006D8: E4 E3 DD 0D 7A 4C 16 76 E2 C4 EC 09 01 00 00 00 40 7F 47 00 0C 06 00 00 ääÝ.zL.väÄi...@EG.....
0000006F0: 0C 16 BE 09 00 00 00 78 E4 DD 0D 30 1C BB 07 00 00 61 00 6C 00 69 00 ..¼.....xäÝ.0.»...a.l.i.
000000708: 62 00 72 00 69 00 00 00 40 91 19 76 01 00 00 00 71 00 00 00 58 00 00 00 b.r.i...@`v....q...X...
000000720: 19 00 00 00 15 00 00 00 00 00 00 00 30 00 00 00 A0 00 00 00 90 01 00 000... ..•....
000000738: 00 00 00 00 58 02 00 00 00 00 00 00 BC 66 F7 09 38 E4 DD 0D A2 91 F7 38X.....¼f÷.8äÝ.¢÷8
000000750: 3B 11 21 B3 74 E4 DD 0D 70 E4 DD 0D 78 E4 DD 0D 54 E4 DD 0D BF 90 F7 38 ;.!³täÝ.päÝ.xäÝ.TäÝ.¿•÷8
000000768: 00 60 5F 00 3B 11 21 B3 74 E4 DD 0D 70 E4 DD 0D 34 FE 15 76 1C FE 15 76 .`_.;.!³täÝ.päÝ.4p.v.p.v
000000780: 64 76 00 08 00 00 00 25 00 00 00 0C 00 00 00 04 00 00 00 28 00 00 00 dv.....%.....(....
000000798: 0C 00 00 00 03 00 00 00 25 00 00 00 0C 00 00 00 02 00 00 00 28 00 00 00%.....(....
0000007B0: 0C 00 00 00 04 00 00 00 12 00 00 00 0C 00 00 00 01 00 00 00 18 00 00 00
0000007C8: 0C 00 00 00 00 00 00 02 6D 00 00 00 10 00 00 00 A5 52 0C F3 01 00 00 00m.....YR.ó....
0000007E0: 54 00 00 00 A0 00 00 00 61 02 00 00 E9 01 00 00 93 04 00 00 59 02 00 00 T... ..a...é...".Y...
0000007F8: 01 00 00 00 CF 2E 87 40 C2 5E 87 40 61 02 00 00 41 02 00 00 0E 00 00 00î.+@Ä^+@a...A.....
000000810: 4C 00 00 00 04 10 00 00 00 00 00 00 00 00 00 61 13 00 00 68 1B 00 00 L.....a...h...
000000828: 68 00 00 00 50 00 72 00 69 00 6D 00 65 00 72 00 61 00 20 00 50 00 E1 00 h...P.r.i.m.e.r.a. .P.á.
000000840: 67 00 69 00 6E 00 61 00 30 00 00 00 20 00 00 00 15 00 00 00 49 00 00 00 g.i.n.a.0.... ..I...
000000858: 2E 00 00 00 20 00 00 00 2C 00 00 00 14 00 00 00 30 00 00 00 2C 00 00 00,.....0.....
000000870: 2B 00 00 00 15 00 00 00 30 00 00 00 2B 00 00 00 18 00 00 00 0C 00 00 00 +.....0...+.....
000000888: 00 00 00 02 54 00 00 00 54 00 00 00 94 04 00 00 E9 01 00 00 43 09 00 00T...T..."...é...C...
0000008A0: 59 02 00 00 01 00 00 00 CF 2E 87 40 C2 5E 87 40 94 04 00 00 41 02 00 00 Y.....î.+@Ä^+@"...A...
0000008B8: 01 00 00 00 4C 00 00 00 04 10 00 00 00 00 00 00 00 00 61 13 00 00L.....a...
0000008D0: 68 1B 00 00 50 00 00 00 20 00 00 00 B0 04 00 00 18 00 00 00 0C 00 00 00 h...P... ..°.....
0000008E8: 00 00 00 02 54 00 00 00 54 00 00 00 44 09 00 00 E9 01 00 00 79 09 00 00T...T...D...é...y...
000000900: 59 02 00 00 01 00 00 00 CF 2E 87 40 C2 5E 87 40 44 09 00 00 41 02 00 00 Y.....î.+@Ä^+@D...A...
000000918: 01 00 00 00 4C 00 00 00 04 10 00 00 00 00 00 00 00 00 61 13 00 00L.....a...
000000930: 68 1B 00 00 50 00 00 00 20 00 00 00 36 00 00 00 18 00 00 00 0C 00 00 00 h...P... ..6.....
000000948: 00 00 00 02 25 00 00 00 0C 00 00 00 0E 00 00 80 0E 00 00 00 14 00 00 00%.....€.....
000000960: 00 00 00 00 10 00 00 00 14 00 00 00 0E 00 00 00 08 00 00 00 20 09 00 00
000000978: 00 00 00 00 0C 00 00 00 24 04 00 00 01 00 00 00 84 00 00 00 61 02 00 00\$......„a...
000000990: E9 01 00 00 CD 04 00 00 59 02 00 00 00 00 00 00 00 00 84 4E 00 00 é...í...Y.....„N..
0000009A8: 80 70 00 00 20 45 4D 46 00 00 01 00 24 04 00 00 1C 00 00 00 02 00 00 00 €p.. EMF....\$......
0000009C0: 0C 00 00 00 6C 00 00 00 00 00 00 00 96 12 00 00 98 1A 00 00 C9 00 00 00l.....-...~...É...
0000009D8: 20 01 00 00 00 00 00 00 00 00 00 00 00 00 00 DC 12 03 00 DC 65 04 00Ü...Üe..
0000009F0: 50 00 72 00 69 00 6E 00 74 00 20 00 74 00 65 00 73 00 74 00 00 00 00 P.r.i.n.t. .t.e.s.t....
000000A08: 25 00 00 00 0C 00 00 00 07 00 00 80 25 00 00 00 0C 00 00 00 00 00 80 %.....€%.....e
000000A20: 25 00 00 00 0C 00 00 00 0E 00 00 80 18 00 00 00 0C 00 00 00 00 00 02 %.....€.....
000000A38: 12 00 00 00 0C 00 00 00 01 00 00 16 00 00 00 0C 00 00 00 18 00 00 00
000000A50: 1B 00 00 00 10 00 00 00 00 00 00 00 00 00 00 0D 00 00 00 10 00 00 00
000000A68: 00 00 00 00 00 00 00 00 62 00 00 00 0C 00 00 00 01 00 00 00 64 00 00 00b.....d...
000000A80: 0C 00 00 00 14 00 00 80 0C 00 00 10 00 00 00 00 00 00 00 00 00 00e.....
000000A98: 6A 00 00 00 10 00 00 14 00 00 00 00 00 00 00 25 00 00 00 0C 00 00 00 j.....%.....
000000AB0: 0E 00 00 80 16 00 00 0C 00 00 18 00 00 00 0C 00 00 00 10 00 00 00 ..€.....
000000AC8: 00 00 00 00 00 00 00 12 00 00 00 0C 00 00 00 01 00 00 00 12 00 00 00
000000AE0: 0C 00 00 00 01 00 00 52 00 00 00 70 01 00 00 01 00 00 00 A4 FF FF FFR...p.....nýÿÿ
000000AF8: 00 00 00 00 00 00 00 00 00 00 00 00 90 01 00 00 00 00 00 04 40 00 22•.....@.""
000000B10: 43 00 61 00 6C 00 69 00 62 00 72 00 69 00 00 00 00 00 00 00 00 00 00 C.a.l.i.b.r.i.....
000000B28: 00
000000B40: 00
000000B58: 00
000000B70: 00øäÝ.
000000B88: 70 E2 DD 0D 70 8B 55 01 0C DF DD 0D 07 61 63 31 70 E2 DD 0D 20 00 00 00 päÝ.p.xU..äÝ..aclpäÝ. ...
000000BA0: 64 DF DD 0D FF FF FF FF 54 E2 DD 0D EC DF DD 0D 52 1B 6C 31 64 DF DD 0D dšÝ.ÿÿÿÿTäÝ.išÝ.R.lldšÝ.
000000BB8: 70 E2 DD 0D 20 00 00 00 FF FF FF FF 6C 2C 54 01 C9 1B 6C 31 01 00 00 00 päÝ. ...ÿÿÿÿl,T.É.ll....
000000BD0: 00 00 00 00 58 02 00 00 25 00 00 00 37 2E 90 01 00 00 00 00 00 00 00X...%...7.~.....
000000BE8: 00 00 00 00 FF 02 00 E0 FF AC 00 40 01 00 00 00 00 00 00 00 9F 01 00 00ÿ..äÿ~.@.....ÿ...
000000C00: 00 00 00 00 43 00 61 00 6C 00 69 00 62 00 72 00 00 00 00 00 00 00 00C.a.l.i.b.r.....
000000C18: 00 00 00 00 00 00 00 00 00 00 94 DF DD 0D 79 2A 66 31 08 00 00 00"
000000C30: FF FF FF DF D0 DF DD 0D 2A 73 64 31 E4 DF DD 0D A0 01 57 32 FC DF DD 0D ÿÿÿßðšÝ.*sd1äšÝ. .w2üšÝ.

```

000000C48: 6C 2C 54 01 90 2B 54 01 64 76 00 08 00 00 00 00 25 00 00 00 0C 00 00 00 1,T.●+T.dv.....%.....
000000C60: 01 00 00 00 18 00 00 00 0C 00 00 00 00 B0 50 02 6D 00 00 00 10 00 00 00 .....°P.m.....
000000C78: A5 52 0C F3 01 00 00 00 54 00 00 00 9C 00 00 00 61 02 00 00 E9 01 00 00 ¥R.ó....T...æ...a...é...
000000C90: 97 04 00 00 59 02 00 00 01 00 00 00 CF 2E 87 40 C2 5E 87 40 61 02 00 00 -...Y.....İ.+@Ã^+@a...
000000CA8: 41 02 00 00 0D 00 00 00 4C 00 00 00 04 10 00 00 00 00 00 00 00 00 00 00 A.....L.....
000000CC0: 61 13 00 00 68 1B 00 00 68 00 00 00 53 00 65 00 67 00 75 00 6E 00 64 00 a...h...h...S.e.g.u.n.d.
000000CD8: 61 00 50 00 E1 00 67 00 69 00 6E 00 61 00 00 00 2A 00 00 00 2E 00 00 00 a.P.á.g.i.n.a...*.....
000000CF0: 2B 00 00 00 30 00 00 00 30 00 00 00 30 00 00 00 2C 00 00 00 30 00 00 00 +...0...0...0...0...
000000D08: 2C 00 00 00 2B 00 00 00 15 00 00 00 30 00 00 00 2C 00 00 00 18 00 00 00 ,...+.....0.....
000000D20: 0C 00 00 00 00 00 00 02 54 00 00 00 54 00 00 00 98 04 00 00 E9 01 00 00 .....T...T...~...é...
000000D38: CD 04 00 00 59 02 00 00 01 00 00 00 CF 2E 87 40 C2 5E 87 40 98 04 00 00 í...Y.....İ.+@Ã^+@~...
000000D50: 41 02 00 00 01 00 00 00 4C 00 00 00 04 10 00 00 00 00 00 00 00 00 00 00 A.....L.....
000000D68: 61 13 00 00 68 1B 00 00 50 00 00 00 20 00 00 00 36 00 00 00 18 00 00 00 a...h...P... ..6.....
000000D80: 0C 00 00 00 00 00 00 02 25 00 00 00 0C 00 00 00 0E 00 00 80 0E 00 00 00 .....%......€.....
000000D98: 14 00 00 00 00 00 00 10 00 00 00 14 00 00 00 0D 00 00 00 08 00 00 00 .....
000000DB0: 2C 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ,.....

```

2.2.5.4.2.1 Análisis EMFSPool/EMF

A continuación se muestra el análisis del fichero de spool generado al imprimir el documento en una impresora HP LaserJet 2605 que utiliza lenguaje EMF.

Pero antes veamos algunos detalles de la estructura del documento:

- Empieza por el Registro de cabecera HEADER_RECORD en la posición 0x00000000. Este registro aparece una única vez al principio del fichero de spool
- El siguiente registro de EMFSPool es un registro EMRI_METAFILE_DATA, Pos. 0x0000004C, que contiene un metarchivo EMF, es decir contiene una página del documento, definida por una colección de registros EMR (EMR: Enhanced Metafile Records).
 - o El contenido del metarchivo comienza por un registro EMR_HEADER Pos. 0x00000054. Podemos analizar el contenido del campo Frame; define un rectángulo cuya esquina inferior izquierda tiene de coordenada (0, 0) y su esquina superior derecha tiene de coordenada (20100, 28800), ambas coordenadas están expresadas en milésimas de cm. Es decir define un área imprimible de 20,1x28,8 cm².
 - o Sigue una colección de registros EMR que definen todo el contenido de la página.
 - o Acaba por un registro EMR_EOF, Pos. 0x000000958, que indica el final del metarchivo.
- A continuación aparece un registro EMFSPool de posición de página EMRI_BW_METAFILE_EXT, Pos. 0x00000096c, para señalar el final de página. Este tipo de registro indica que el contenido de la página era en b/n.
- De nuevo se vuelve a definir una nueva página con un registro EMFSPool tipo EMRI_METAFILE_DATA, Pos. 0x00000097c, que contiene otro metarchivo EMF con una estructura similar a la vista:
 - o Un registro de cabecera EMR_HEADER, Pos. 0x000000984. El campo frame define un rectángulo de las mismas dimensiones que la cabecera de la página anterior.
 - o Colección de registros EMR.
 - o Finalizar con un registro EMR_EOF, Pos. 0x000000d94. Indica el final del metarchivo.
- El fichero de Spool acaba con otro registro EMFSPool de posición de página EMRI_METAFILE_EXT, Pos. 0x000000da8. A diferencia del anterior este indica que la página tiene contenido en color.

En resumen el documento consta de 2 páginas de las que sólo una tiene contenido en color. Sin más se muestra el log con el análisis del fichero de spool generado desde el servicio Monitor de Impresión desarrollado en este TFG gracias a la librería log4net.

```
>>>> Fichero de Spool para analisis EMF: D:\SpoolImpresora\Test\Prueba\EjemploEMF.SPL
Tamaño de fichero SPL para análisis EMF: 3512 bytes.
Analisis mediante lenguaje EMF del ficheros de Spool: D:\SpoolImpresora\Test\Prueba\EjemploEMF.SP
Posición: (0x00000000) registro EMF: HEADER_RECORD Tamaño: 0x0000004C.
Posición siguiente registro EMFSPOOL: (0x0000004C)
Posición: (0x0000004C) registro EMF: EMRI_METAFILE_DATA Tamaño: 0x00000918.
Posición 0x00000054 EMR_HEADER. Tamaño registro: 0x00000084
      0x00000078      Frame:      (0, 0), (20100, 28800)
Posición 0x000000D8 EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x000000E4 EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x000000F0 EMR_EXTCREATEFONTINDIRECTW. Tamaño registro: 0x00000170
Posición 0x00000260 EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x0000026C EMR_SETTEXTALIGN. Tamaño registro: 0x0000000C
Posición 0x00000278 EMR_MOVETOEX. Tamaño registro: 0x00000010
Posición 0x00000288 EMR_SETBRUSHORGE. Tamaño registro: 0x00000010
Posición 0x00000298 EMR_SETICMMode. Tamaño registro: 0x0000000C
Posición 0x000002A4 EMR_SETCOLORSPACE. Tamaño registro: 0x0000000C
Posición 0x000002B0 EMR_ExtEscape. Tamaño registro: 0x00000010
Posición 0x000002C0 EMR_SETVIEWPORTORGE. Tamaño registro: 0x00000010
Posición 0x000002D0 EMR_ExtEscape. Tamaño registro: 0x00000010
Posición 0x000002E0 EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x000002EC EMR_SETTEXTALIGN. Tamaño registro: 0x0000000C
Posición 0x000002F8 EMR_SETVIEWPORTORGE. Tamaño registro: 0x00000010
Posición 0x00000308 EMR_SETBKMODE. Tamaño registro: 0x0000000C
Posición 0x00000314 EMR_EXTCREATEFONTINDIRECTW. Tamaño registro: 0x00000170
Posición 0x00000344 EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x00000490 EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x0000049C EMR_EXTCREATEFONTINDIRECTW. Tamaño registro: 0x00000170
Posición 0x0000060C EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x00000618 EMR_EXTCREATEFONTINDIRECTW. Tamaño registro: 0x00000170
Posición 0x00000788 EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x00000794 EMR_DELETEOBJECT. Tamaño registro: 0x0000000C
Posición 0x000007A0 EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x000007AC EMR_DELETEOBJECT. Tamaño registro: 0x0000000C
Posición 0x000007B8 EMR_SETBKMODE. Tamaño registro: 0x0000000C
Posición 0x000007C4 EMR_SETTEXTCOLOR. Tamaño registro: 0x0000000C
Posición 0x000007D0 EMR_FORCEUFIMAPPING. Tamaño registro: 0x00000010
Posición 0x000007E0 EMR_ExtTextOutW. Tamaño registro: 0x000000A0
Posición 0x00000880 EMR_SETTEXTCOLOR. Tamaño registro: 0x0000000C
Posición 0x0000088C EMR_ExtTextOutW. Tamaño registro: 0x00000054
Posición 0x000008E0 EMR_SETTEXTCOLOR. Tamaño registro: 0x0000000C
Posición 0x000008EC EMR_ExtTextOutW. Tamaño registro: 0x00000054
Posición 0x00000940 EMR_SETTEXTCOLOR. Tamaño registro: 0x0000000C
Posición 0x0000094C EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x00000958 EMR_EOF. Tamaño registro: 0x00000014
Posición siguiente registro EMFSPOOL: (0x0000096C)
Posición: (0x0000096C) registro EMF: EMRI_BW_METAFILE_EXT Tamaño: 0x00000008.
Posición siguiente registro EMFSPOOL: (0x0000097C)
Posición: (0x0000097C) registro EMF: EMRI_METAFILE_DATA Tamaño: 0x00000424.
Posición 0x00000984 EMR_HEADER. Tamaño registro: 0x00000084
      0x000009A8      Frame:      (0, 0), (20100, 28800)
Posición 0x00000A08 EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x00000A14 EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x00000A20 EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x00000A2C EMR_SETTEXTCOLOR. Tamaño registro: 0x0000000C
Posición 0x00000A38 EMR_SETBKMODE. Tamaño registro: 0x0000000C
Posición 0x00000A44 EMR_SETTEXTALIGN. Tamaño registro: 0x0000000C
Posición 0x00000A50 EMR_MOVETOEX. Tamaño registro: 0x00000010
Posición 0x00000A60 EMR_SETBRUSHORGE. Tamaño registro: 0x00000010
Posición 0x00000A70 EMR_SETICMMode. Tamaño registro: 0x0000000C
Posición 0x00000A7C EMR_SETCOLORSPACE. Tamaño registro: 0x0000000C
Posición 0x00000A88 EMR_SETVIEWPORTORGE. Tamaño registro: 0x00000010
Posición 0x00000A98 EMR_ExtEscape. Tamaño registro: 0x00000010
Posición 0x00000AA8 EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x00000AB4 EMR_SETTEXTALIGN. Tamaño registro: 0x0000000C
Posición 0x00000AC0 EMR_SETVIEWPORTORGE. Tamaño registro: 0x00000010
```

Posición 0x00000AD0 EMR_SETBKMODE. Tamaño registro: 0x0000000C
Posición 0x00000ADC EMR_SETBKMODE. Tamaño registro: 0x0000000C
Posición 0x00000AE8 EMR_EXTCREATEFONTINDIRECTW. Tamaño registro: 0x00000170
Posición 0x00000C58 EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x00000C64 EMR_SETTEXTCOLOR. Tamaño registro: 0x0000000C
Posición 0x00000C70 EMR_FORCEUFIMAPPING. Tamaño registro: 0x00000010
Posición 0x00000C80 EMR_EXTTEXTOUTW. Tamaño registro: 0x0000009C
Posición 0x00000D1C EMR_SETTEXTCOLOR. Tamaño registro: 0x0000000C
Posición 0x00000D28 EMR_EXTTEXTOUTW. Tamaño registro: 0x00000054
Posición 0x00000D7C EMR_SETTEXTCOLOR. Tamaño registro: 0x0000000C
Posición 0x00000D88 EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
Posición 0x00000D94 EMR_EOF. Tamaño registro: 0x00000014
Posicion siguiente registro EMFSPOOL: (0x00000DA8)
Posición: (0x00000DA8) registro EMF: EMRI_METAFILE_EXT Tamaño: 0x00000008.
Posicion siguiente registro EMFSPOOL: (0x00000DB8)
Cerramos el fichero EMF

Resumen Tags con el número de ocurrencias.

EMR_DELETEOBJECT: 2
EMR_EOF: 2
EMR_EXTCREATEFONTINDIRECTW: 5
EMR_EXTESCAPE: 3
EMR_EXTTEXTOUTW: 5
EMR_FORCEUFIMAPPING: 2
EMR_HEADER: 2
EMR_MOVETOEX: 2
EMR_SELECTOBJECT: 16
EMR_SETBKMODE: 5
EMR_SETBRUSHORGE: 2
EMR_SETCOLORSPACE: 2
EMR_SETICMMode: 2
EMR_SETTEXTALIGN: 4
EMR_SETTEXTCOLOR: 8
EMR_SETVIEWPORTORGE: 4
EMRI_BW_METAFILE_EXT: 1
EMRI_METAFILE_DATA: 2
EMRI_METAFILE_EXT: 1
HEADER_RECORD: 1

Total: 71

FIN Analisis EMF

3. METODOLOGÍA E INFRAESTRUCTURA UTILIZADA PARA EL DESARROLLO DEL TFG

3.1 Equipos y dispositivos de impresión.

Este trabajo se ha desarrollado utilizando la infraestructura del Servicio de impresión ofrecido por el STIC. Estos son los equipos y dispositivos de impresión utilizados:

- Un ploter HP DesignJet Z3200 Photo conectado localmente a un PC con Sistema Windows 7 Enterprise x64. Los usuarios tienen que imprimir de forma local a través del PC.
- UN ploter HP DesignJet 800 (42 in) conectado localmente a un PC con Sistema Windows 7 Enterprise x64. Los usuarios tienen que imprimir de forma local a través del PC.
- Una impresora JP LaserJet 5500 (color) conectada de forma local al servidor Imprenta con sistema operativo Windows 2003 Server Estándar Edition. Esta impresora está compartida a todos los usuarios que tienen cuenta en WUAL (red de dominio Microsoft en la UAL). Los usuarios lanzan sus trabajos de impresión desde los PCs de la sala de usuarios del STIC validándose con su usuario de dominio WUAL; estos PCs ya tiene instalados los drivers de la impresora compartida por lo que no tienen que realizar ninguna acción adicional para poder lanzar sus trabajos (salvo iniciar sesión en el dominio microsoft WUAL). Además la impresora tiene una tarjeta de red (y está conectada a la red) por lo que es posible configurarla como impresora de red desde cualquier PC con IP perteneciente a la misma subred.

También he utilizado las siguientes impresoras de red del STIC:

- HP LaseJet 2605 (color)
- Samsung SCX-5835_5935 (b/n)

Se ha utilizado un servidor con Windows 2008 Server R2 Enterprise para alojar la capa de acceso a datos CAD.

Un S.G.B.D Oracle con la BD utilizada para almacenar las propiedades de los trabajos de impresión.

3.2 Herramientas

3.2.1 Desarrollo SW y BD.

Visual Studio(Alvorsen, 2012): Es un entorno de desarrollo completo para gestión de proyectos. Aplicado a la implementación del Monitor de Impresión facilita la utilización del framework .NET y las APIs del sistema de impresión, también facilita el desarrollo de Servicios de Sistema y su correspondiente instalador, así como servicios WCF. También permite desarrollar aplicaciones de escritorio Windows, aplicaciones ASP.NET. Incluye varios lenguajes de programación entre ellos C# y Visual Basic. También permite agregar de forma manual componentes de terceros como ODP.NET para acceso a BD Oracle; o utilizando el gestor de paquetes NuGet integrado en Visual Studio agregar componentes de terceros como Log4net para generar logs.

De los lenguajes soportados por Visual Studio voy a utilizar C#(Ferguson, Patterson, Beres, Boutquin, & Gupta, n.d.)(Types, 1990) por que me resulta simple, pero a la vez potente y elegante; es un lenguaje ampliamente utilizado por lo que se pueden encontrar muchos recursos además de los ofrecidos a través del MSDN de Microsoft.

Por otra parte la Universidad de Almería está suscrita al programa DreamSpark de Microsoft, por el que se proporciona acceso gratuito a herramientas profesionales de diseño y desarrollo para estudiantes y profesorado entre ellas Visual Studio.

Por tanto considero que es una herramienta idónea para desarrollar este TFG.

Toad for Oracle 12.5: Herramienta de desarrollo SQL y administración de BD Oracle (también disponibles para otras BD como MySql y Microsoft SQL Server)

Toad Data Modeler 5.2: Herramienta que permite diseñar esquemas de BD y generar el código SQL para crearlas.

3.2.2 Utilidades

XVI32: Es un editor hexadecimal de ficheros que nos permite, entre otras cosas, analizar el contenido binario de los ficheros de Spool de los trabajos de impresión.

Visor de Eventos de Windows: Con esta aplicación podemos supervisar los eventos que se producen en el sistema, en concreto nos permite supervisar los eventos del Servicio de Impresión de Windows.

Log4net: Módulo para generación de logs disponibles para Visual Studio

Paraphernalia PCL: Utilidad para análisis de ficheros con contenido PCL, JPL, PCLXL

Notepad++: Editor de texto y de código fuente que admite un gran número de lenguajes de programación.

PDFCreator: Se instala como controlador de impresora y genera como salida un documento PDF. Resulta muy útil en este proyecto pues permite hacer numerosas pruebas con el Servidor de Impresión sin tener que gastar papel.

Doxygen: Generador de documentación de código C#. La documentación se genera a partir de los comentarios introducidos en el código de la aplicación.

Administración de Impresión (Componente Windows): Es una herramienta gráfica del sistema que permite administrar el Servicio de impresión Windows.

3.2.3 Planificación.

Visual Studio Team Foundation Server (TFS) 2015: Es una herramienta que facilita el trabajo en grupo. Da buen soporte a metodologías ágiles como SCRUM permitiendo definir iteraciones, dividir Funcionalidades en Elementos de trabajo y Tareas, utilizar un tablero Kanban(Figuerola, 2004) digital, Visualizar diagramas de progreso Burndown, etc. Entre sus funciones interesantes está el control de versiones y compilación continua. Esta herramienta complementa a Visual Studio 2015.

Microsoft Project 2013: Según Wikipedia *“Microsoft Project (o MSP) es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.”*

3.2.4 BD

Crearé la BD de la aplicación Monitor Impresión en un esquema de BD Oracle existente, en el que ya existe una BD donde se almacenan cuentas de usuario, como las del dominio Microsoft WUAL. De esta manera en el trabajo posterior al de este TFG dispondremos de la información necesaria en un mismo esquema para calcular costes de impresión por usuario.

3.2.5 Ofimáticas

Office 2010: Suite ofimática de Microsoft. En concreto el procesador de texto Microsoft Word 2010 para realizar la documentación de este TFG y el SGBD Microsoft Access para realizar algunas tareas durante el desarrollo del TFG.

Microsoft Visio 2013: Utilizado para realizar parte de los diagramas presentados en las ilustraciones. Permite realizar diagramas de oficina, UML, de flujo de programas, de BD, etc.

3.2.6 Servidor de aplicaciones

Internet Information Server (IIS) 7.5: Utilizado como servidor de aplicaciones donde poder alojar servicios WCF. También admite sitios web desarrollados en ASP.NET así como control de acceso de usuarios.

3.3 Metodologías.

Para cada fase del proyecto he utilizado la metodología, o marco de trabajo, que he considerado más adecuado.

3.3.1 Modelo de Prototipos

Se pueden consultar las referencias (Pressman, 2002)(Inteco, 2009).

Este modelo es útil para delimitar el alcance de un proyecto, dónde sólo están definidos unos objetivos generales pero no están definidos de forma clara los requisitos de entrada, de proceso, de salida.

En estas situaciones es esencial la construcción de prototipos que permiten una mayor comprensión de las dificultades que puede entrañar la realización de un proyecto, identificando las funcionalidades y requisitos clave (entrada, salida, proceso).

He utilizado este modelo en la primera fase donde no estaba claro cual era el proceso necesario para capturar los detalles de los trabajos de impresión ni la dificultad que conlleva.

La siguiente ilustración muestra la idea de este modelo:



Ilustración 14. Modelo Prototipo. Fuente (Inteco, 2009)

3.3.2 Modelo Incremental

Se pueden consultar las referencias (Pressman, 2002)(Inteco, 2009).

Este modelo combina el desarrollo tradicional en cascada con la construcción interactiva de prototipos. Podría verse como la siguiente etapa en el modelo de de prototipos donde en lugar de desechar los prototipos se reutilizan dotándolos de plena funcionalidad.

Con esta idea he utilizado este modelo de trabajo en la segunda fase, una vez visto que el resultado obtenido con los prototipos era satisfactorio. El objetivo ha sido obtener un producto totalmente funcional para poder presentarlo a los usuarios para ir mejorándolo en los siguientes incrementos.

En la siguiente ilustración se esquematiza la idea de este modelo:

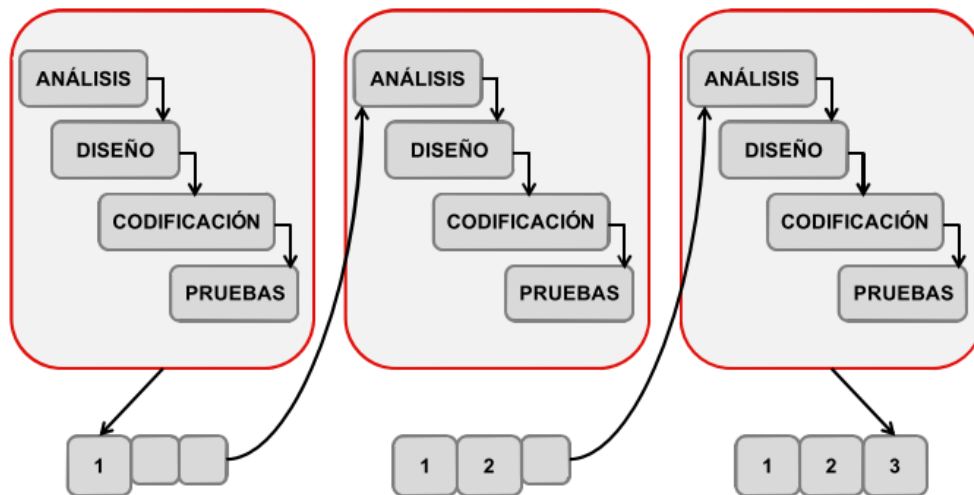


Ilustración 15. Modelo Incremental. Fuente (Inteco, 2009)

3.3.3 Scrum

(Kniberg, 2007)(Schwaber & Sutherland, 2013)

Aunque en la 2ª fase de desarrollo del proyecto se dispone de un producto funcional, todavía hay algunos detalles que resolver para poder considerarlo como un producto final, estos son:

- En primer lugar las pruebas realizadas no son lo suficientemente fiables y en los trabajos capturados hay información que habría que contrastar en un entorno de producción.
- No hay una implicación suficiente de los usuarios finales al que va destinado el resultado de este trabajo para la revisión de requisitos y aprobación del trabajo.
- Poder decidir que producto o partes del producto están finalizadas.

Por ello es necesario utilizar un nuevo marco de trabajo que dé respuesta a estos problemas. Las metodologías ágiles, en concreto Scrum, van uno paso más allá y dan respuesta a estos inconvenientes.

Aunque Scrum no es nuevo, si ha sido novedoso para mí, por eso lo describo con mayor detalle en los siguientes apartados.

3.3.3.1 ¿Por qué utilizar una metodología ágil?

(Cadavid, Fernández Martínez, & Morales Vélez, 2013)

Las metodologías tradicionales de ingeniería del SW son heredadas de las metodologías utilizadas en las ingenierías destinadas a obtener productos físicos en las que los proyectos se organizan por fases (Especificación de requisitos, Análisis, Diseño, Pruebas, Implantación y Mantenimiento). Hasta que no termina una fase no comienza la siguiente siendo muy costoso introducir cambios en fases iniciales a medida que avanza el proyecto. Son metodologías poco flexibles y hasta que no finalizan todas las fases no se obtiene un producto.

Las metodologías tradicionales son adecuadas en proyectos de desarrollo de SW en los que se tiene muy claro que se necesita con requisitos y plazos muy cerrados. Tiene sentido para grupos de trabajo grandes. Origina una extensa documentación. Muchos formalismos.

Las metodologías ágiles son adecuadas en aquellos desarrollos SW donde es necesario disponer de algunas funcionalidades cuanto antes, donde los requerimientos iniciales cambian con frecuencia a medida que avanza el proyecto por necesidades del cliente o por necesidades descubiertas durante el desarrollo y que no se habían tenido en cuenta.

Las metodologías ágiles son adecuadas para el desarrollo del SW que sigue un ciclo iterativo e incremental. Al finalizar un ciclo se tiene un producto funcionando y probado. En cada ciclo se van añadiendo funcionalidades. Es adecuada para grupos de trabajos reducidos e involucrados.

Para el trabajo que se quiere realizar en este TFG, en concreto la tercera fase del mismo, encaja mejor una metodología ágil. Durante el desarrollo del proyecto es muy probable que surjan problemas por lo que haya que reconsiderar el trabajo precedente y los requisitos iniciales con frecuencia.

Existen cantidad de metodologías ágiles. Entre las más populares están Scrum, XP, RUP, Kanban, Scrumban. Estas se pueden adaptar según el entorno en que se aplica incluso puede ser recomendable usar una mezcla de ellas.

3.3.3.1.1 Manifiesto ágil("Manifiesto por el Desarrollo Ágil de Software," n.d.)

"Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

- ✓ **Individuos e interacciones** sobre procesos y herramientas
- ✓ **Software funcionando** sobre documentación extensiva
- ✓ **Colaboración con el cliente** sobre negociación contractual
- ✓ **Respuesta ante el cambio** sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda."

3.3.3.1.1.1 Principios

"Seguimos estos principios:

Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.

El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

El software funcionando es la medida principal de progreso.

Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.

A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.”

3.3.3.2 Scrum

Para el desarrollo de la 3ª fase del TFG voy a utilizar Scrum por los siguientes motivos:

- ✓ Es una metodología ágil por lo que se basa en desarrollos iterativos e incrementales.
- ✓ Simplicidad de la metodología: Para evitar rechazo por parte del equipo implicado.
- ✓ Ampliamente utilizada.
- ✓ Facilidad de implantación
- ✓ Idea principal es ponerse a trabajar desde el primer momento y sacar frutos de ese trabajo muy pronto
- ✓ Evitar exceso de documentación
- ✓ Fomenta la motivación del equipo

En Scrum se diferencian tres tipos de elementos Actores, Herramientas y acciones.

Actores o Roles:

- ✓ Product Owner (cliente).
- ✓ Scrum Master.
- ✓ Scrum Team

Herramientas:

- ✓ Product Backlog
- ✓ Spring Backlog
- ✓ Gráficos de trabajo pendiente (Burndown)
- ✓ Estimación con planning poker

Reuniones:

- ✓ Planificación del proyecto
- ✓ Planificación Sprint
- ✓ Revisión Sprint
- ✓ Retrospectiva Sprint
- ✓ Reunión diaria de Scrum

3.3.3.2.1 Actores

3.3.3.2.1.1 Product Owner (Cliente)

- ✓ Es el cliente, representa a todas las personas interesadas en los resultados del proyecto.
- ✓ Tiene autoridad y toma decisiones
- ✓ Define los objetivos y los prioriza
- ✓ Es el propietario del Product Backlog

3.3.3.2.1.2 *Scrum Master (Director)*

- ✓ Lidera el equipo. Sólo uno por equipo.
- ✓ Trabaja junto con el equipo.
- ✓ Vela por que se sigan las reglas y proceso de Scrum adaptándolas a la organización.
- ✓ Facilita la colaboración del equipo con el cliente asegurándose de que esté preparada la lista de requisitos priorizada antes de empezar una nueva iteración.
- ✓ Facilita las distintas reuniones de Scrum.
- ✓ Proteger al equipo de interrupciones para que el equipo pueda mantener la productividad.
- ✓ Asegurar que se consiguen los requisitos con la suficiente calidad.

3.3.3.2.1.3 *Scrum Team (Equipo)*

- ✓ Es el grupo de personas encargadas de desarrollar el producto.
- ✓ Comparte responsabilidad del trabajo y su calidad.
- ✓ Es un equipo auto gestionado que realiza de manera conjunta las siguientes actividades:
 - Seleccionar los requisitos que se comprometen a realizar en cada iteración
 - Estimar la complejidad de cada requisito, las tareas necesarias para realizarlo y el esfuerzo para completar cada requisito. Los miembros del equipo se asignan a las tareas.
- ✓ Trabajar de manera conjunta durante cada iteración.
- ✓ Cuando finaliza la iteración mostrar al cliente los requisitos completados.
- ✓ El equipo es multidisciplinar. Tienen que depender lo menos posible de personas ajenas al equipo.
- ✓ Los miembros del equipo deben trabajar en la misma localización para facilitar la comunicación.
- ✓ El equipo debe ser estable.
- ✓ Estiman la complejidad de los requisitos priorizados el backlog.
- ✓ La experiencia muestra que el tamaño ideal esta entre 5 y 9 personas. En equipos de menos de 5, ante cualquier eventualidad de algún miembro, el tiempo de ejecución de tareas en el sprint se ve comprometido. En equipos de más de 9 miembros la comunicación y colaboración se hace más difícil.

Scrum sistematiza la colaboración dentro del equipo mediante las reuniones de:

- ✓ planificación del sprint.
- ✓ Reunión diaria del sprint.
- ✓ Reunión retrospectiva.

3.3.3.2.2 *Herramientas*

3.3.3.2.2.1 *Product Backlog*

Consiste en una lista priorizada de requerimientos (Historias de usuario o funcionalidades) definidos a alto nivel y priorizados por el cliente (Product Owner) . Es una lista dinámica que puede ser repriorizada al comiendo de cada sprint. Pueden tener los siguientes campos.

Identificador: Identificador único. Autoincremental.

Historia: Descripción corta pero suficientemente clara de una funcionalidad.

Importancia: Valor que indica importancia relativa con respecto al resto de historias. Cuanto mayor es el valor mayor importancia.

Estimación: Valoración inicial que hace el equipo del trabajo necesario para realizarla.

Notas: Cualquier otra información, aclaración. Muy breve.

Como probarlo: Especificación simple de un test o varios test para probar la historia.

Id	Historia	Imp.	Est.	Notas	Como probarlo
1	Suma	20	5	Sólo valores positivos	Probar Introducir valores numéricos. Probar con uno de los valores negativos. Probar que un valor sea una letra.

3.3.3.2.2.2 Spring Backlog

Es un subconjunto de requerimientos del Product Backlog. Estos requerimientos se desglosan a más bajo nivel en tareas. El equipo hace estimaciones de las tareas a realizar y elige el orden en que las van realizando, estas estimaciones son actualizadas diariamente. Cualquier miembro del equipo puede modificar el Sprint Backlog.

3.3.3.2.2.3 Gráfico de trabajo pendiente (Burndown)

Este gráfico muestra el progreso del trabajo durante el sprint. En la siguiente ilustración se muestra un ejemplo:

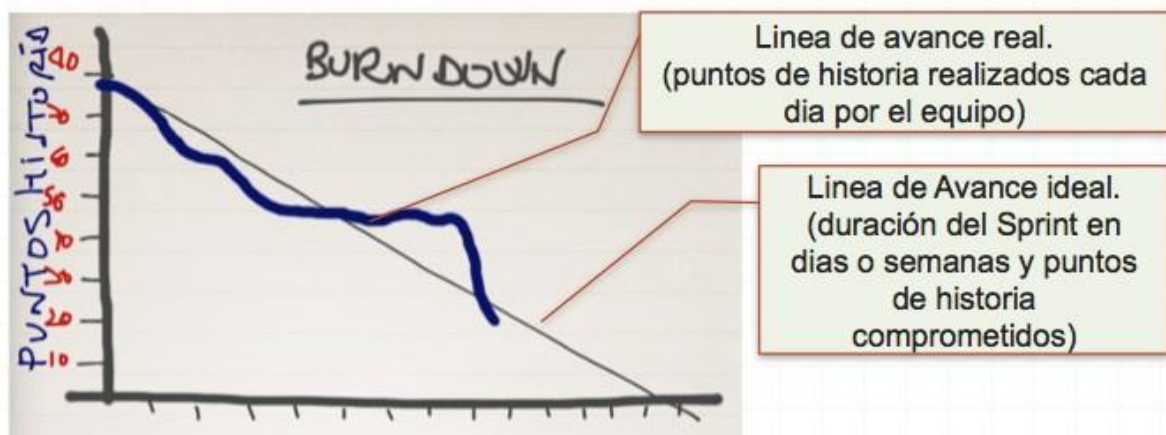


Ilustración 16 Ejemplo Diagrama Burndown.

Fuente: <http://www.gestiondeproyectosit.es/blogit/2012/12/metricas-agiles-ii/>

En el eje Y se representa el trabajo total necesario para realizar el sprint. Como unidades se puede tomar horas de trabajo, puntos de trabajo (idealmente representa la jornada de trabajo de una persona a tiempo completo), o cualquier otra medida de trabajo.

En el eje X se representa la duración del sprint en días.

Con esta gráfica se puede visualizar de un vistazo el progreso del Sprint de modo que ayuda a decidir si es necesario reestimar tareas, adición nuevas tareas, etc.

También permite realizar simulaciones para ver como varían las fechas de entrega cuando se añaden o quitan requisitos o se añade otro equipo.

Para construir la gráfica burndown hay que saber cuál va a ser la velocidad de trabajo real del equipo durante la ejecución del proyecto, es decir cuántas unidades de trabajo reales pueden realizar el equipo cada día. Así la cantidad de trabajo total que puede realizar el equipo durante el sprint se calcula multiplicando la velocidad del equipo por el número de días que dura el sprint.

Por ejemplo suponemos que un equipo tiene una velocidad de trabajo de 8 puntos/día. Si la duración del sprint dura 10 días reales (2 semanas, cada semana 5 días de trabajo). Podemos calcular el trabajo total que se puede llevar a cabo durante el sprint.

Trabajo total = 8 puntos/día * 10 día = 80 puntos de trabajo.

3.3.3.2.3 Reuniones:

3.3.3.2.3.1 Planificación del proyecto

Se realiza al comienzo del proyecto y participan todos los actores: cliente, director y equipo.

En esta reunión se crea el Product Backlog con los requerimientos del proyecto, qué se quiere hacer a alto nivel. Esta lista se prioriza.

Se discute sobre los costes, riesgos, fechas y estimaciones y se realiza un diseño a muy alto nivel. También se llega a un acuerdo sobre el ritmo o duración más conveniente que deben tener los ciclos o sprint, una vez fijada se debería mantener inamovible durante la duración del proyecto.

3.3.3.2.3.2 Planificación Sprint

En esta reunión interviene el director y el equipo. También debe estar disponible el cliente (teléfono, mail).

El cliente presenta el Product Backlog priorizado y propone los requerimientos a desarrollar. El equipo examina la lista preguntando al cliente las dudas que surgen y selecciona los requisitos que se compromete a completar durante el sprint.

Con el compromiso adquirido el equipo planifica el sprint, elabora la táctica para conseguir el mejor resultado con el mínimo esfuerzo. Es el propio equipo quien organiza su trabajo.

El equipo elabora el Sprint Backlog que consiste en las tareas necesarias para conseguir cada requisito. Realizan una estimación del esfuerzo necesario para cada tarea y cada miembro se autoasigna las tareas que puede realizar.

La duración de esta reunión no debería durar más de una jornada.

3.3.3.2.3.3 Reunión diaria de Scrum

Participa el director y el equipo, la duración no debería ser superior a 15 minutos. Es aconsejable que sea la primera actividad del día. Se recomienda hacerla de pie en el mismo sitio.

Tiene varios objetivos:

- ✓ Identificar los problemas y obstáculos para poder resolverlos cuanto antes.

- ✓ Facilitar la transferencia de información y colaboración para aumentar la productividad del equipo.
- ✓ Cada miembro inspecciona el trabajo del resto así puede hacer adaptaciones necesarias para poder cumplir con el compromiso conjunto.
- ✓ El equipo cuenta con una lista de tareas de la iteración en la que se actualiza el estado y el esfuerzo restante para cada tarea.
- ✓ Se actualiza la gráfica burndown.

Para facilitar la consecución de los objetivos los miembros del equipo deberían responder a las siguientes preguntas:

- ✓ ¿Qué hice desde la última reunión diaria?
- ✓ ¿Qué voy a hacer hasta la próxima reunión diaria?
- ✓ ¿Qué dificultades me he encontrado?

3.3.3.2.3.4 *Revisión Sprint*

Se presenta una demo del producto al cliente para que vea las nuevas funcionalidades. Participan todos los actores.

La duración máxima de la reunión debería ser de 2 a 4 horas; en ella se debería:

Presentar el objetivo del Sprint

Presentar las nuevas funcionalidades al cliente de la forma más real y cercana posible, preparadas para ser entregadas con el mínimo esfuerzo.

El cliente anota mejoras, observaciones y cambios solicitados incorporándolos al Product Backlog y replanificando el proyecto si es necesario.

3.3.3.2.3.5 *Retrospectiva Sprint*

Esta reunión se realiza al finalizar el sprint. En ella participan todos los actores y su duración debería ser de entre 1 hora y un máximo de 3.

Se analiza la manera de trabajar durante el sprint para identificar qué cosas han funcionado bien, cuales han funcionado mal, cuales hay que mejorar.

El objetivo es mejorar de forma continua la productividad y el aprendizaje del equipo de forma sistemática.

3.3.3.2.4 Engranando todas las piezas. Actividades.

En la siguiente ilustración se ve como encajan todos los elementos de SCRUM: Actores, herramientas y reuniones.

The Agile: Scrum Framework at a glance

Información de los ejecutivos,
el equipo, los implicados,
los clientes, los usuarios, etc.

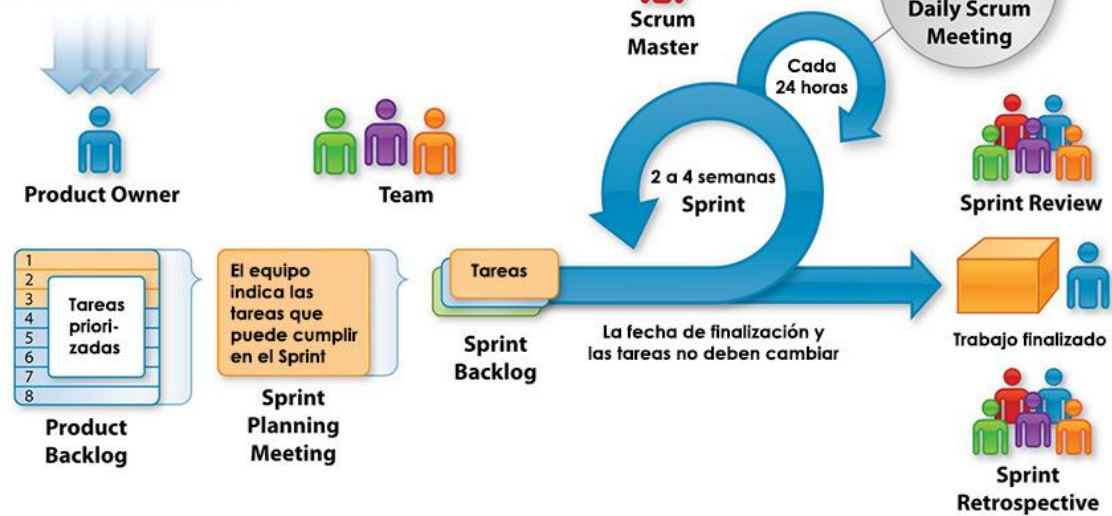


Ilustración 17 Modelo Scrum. Fuente: <http://queondara.readthedocs.org/en/latest/Scrum.html>

En Scrum un proyecto se lleva a cabo en bloques temporales o sprints (ciclos o iteraciones).

En la reunión de planificación del proyecto el cliente, con la ayuda del director y equipo de proyecto, confecciona el Product Backlog; consiste en una lista de requerimientos priorizada. Estos requerimientos se definen, evitando en lo posible tecnicismos, a modo de historias de usuario que suelen corresponderse a unidades funcionales que pueden presentarse a los clientes como elementos acabados.

Se decide cual va a ser la duración del sprint o ritmo de ciclos de proyecto, que debería ser de entre 2 y 4 semanas de duración. Una vez fijada la duración del sprint se mantiene hasta finalizar el proyecto. Atendiendo a su valor y coste se hace una primera división de los requerimientos en iteraciones y entregas.

Cada Sprint comienza con la reunión de planificación del sprint en la que se genera el Sprint Backlog que consiste en una lista de tareas a realizar durante el sprint.

Diariamente los miembros del equipo van actualizando el Sprint Backlog para que todos los implicados en el proyecto puedan conocer el avance del sprint, esto se puede reflejar de forma visual en la gráfica BurnDown.

En la reunión diaria de Scrum se realiza una puesta en común, donde se intercambia información, experiencias, problemas encontrados, etc. que ayuda a avanzar en el Sprint.

Al finalizar el Sprint se reúnen todos los actores en la reunión de revisión del sprint para mostrarle al cliente las nuevas funcionalidades.

Tras esto se hace una reunión retrospectiva identificando posibles mejoras, errores a tener en cuenta en el próximo sprint. Si es necesario se modificarán o incluirán nuevos requerimientos en el product Backlog.

En ocasiones excepcionales un sprint puede acabar antes de tiempo, porque se han finalizado con éxito todas las tareas antes de finalizar el ciclo o bien por que el cliente o el equipo solicita una terminación anormal debido a que el contexto de proyecto ha cambiado sustancialmente y no es posible esperar a finalizar la iteración para aplicar cambios o porque el equipo ve imposible cumplir con el compromiso adquirido.

4. DESARROLLO DEL TFG

Como ya se ha visto, en la introducción de este documento, el desarrollo de este TFG se divide en 3 fases bien diferenciadas en el tiempo y también en los objetivos a conseguir en cada una de ellas. Sin embargo una vez finalizadas las 3 fases presento, para mayor claridad, una vista del desarrollo del proyecto según el ciclo de vida clásico: Análisis, diseño, codificación, pruebas e implantación; con independencia de la fase en la que se ha realizado.

4.1 Análisis del sistema

4.1.1 Identificación de usuarios y componentes con los que interactúa el Servicio Monitor de impresión.

El producto SW desarrollado en este TFG es un Servicio Windows, al que hemos denominado Monitor de Impresión, interactúa con los siguientes componentes de sistema:

- ✓ **Servidor de Impresión Windows.**
- ✓ **Registro de eventos de Windows.**
- ✓ **Sistema de archivos de Windows.**
- ✓ **S.G.B.D.**

La única interacción con usuarios, indirecta, se produce a través del envío de notificaciones a través de email y la generación de logs. Bajo este punto de vista consideramos estos usuarios:

Operador: Recibe por email las propiedades de los trabajos de impresión capturados. Ofrecen asistencia a los usuarios del Servicio de impresión ofrecido por el STIC.

Desarrollador: Recibe emails con los errores de funcionamiento del Monitor de Impresión y supervisa los logs de actividades generados por el Servicio Monitor de Impresión. Es el desarrollador y responsable del mantenimiento del Servicio Monitor de Impresión.

4.1.2 Requisitos SW

Se va a utilizar este formato de ficha para especificar los requisitos:

Requisito <Identificador>		Nombre <Nombre>	
Prioridad	FASE 1	FASE 2	FASE 3
Descripción			

Tabla 6. Plantilla requisito

4.1.2.1 Requisitos Funcionales

Requisito RF-1		Nombre: Capturar evento impresión	
Prioridad	FASE 1 Requerido	FASE 2 Requerido	FASE 3 Requerido
Descripción	Detectar cuando se imprime un documento.		

Tabla 7. Requisito RF-1

Requisito RF-2		Nombre: Capturar evento creación fichero Spool	
Prioridad	FASE 1 Opcional	FASE 2 Deseable	FASE 3 Requerido
Descripción	Detectar cuando se crea un fichero de Spool.		

Tabla 8. Requisito RF-2

Requisito RF-3		Nombre: Capturar detalles de trabajo de impresión	
Prioridad	FASE 1 Requerido	FASE 2 Requerido	FASE 3 Requerido
Descripción	Obtener todos los detalles de impresión utilizando los métodos necesarios: <ul style="list-style-type: none"> - Eventos Impresión - Namespace de .NET System.Printing - API Sistema impresión - Ficheros Spool 		

Tabla 9. Requisito RF-3

Requisito RF-4		Nombre: Guardar log de actividad	
Prioridad	FASE 1 Deseable	FASE 2 Requerido	FASE 3 Requerido
Descripción	Generar un log de la actividad del Servicio Monitor Impresión y Capa CAD (acceso a datos).		

Tabla 10. Requisito RF-4

Requisito RF-5		Nombre: Recuperación Trabajos impresión frente a Error	
Prioridad	FASE 1 Opcional	FASE 2 Deseable	FASE 3 Requerido
Descripción	Ante un error imprevisto en la captura de un trabajo de impresión tener mecanismos para recuperar los detalles de los trabajos de impresión: <ul style="list-style-type: none"> - Guardar una copia de los ficheros de Spool con los trabajos de impresión. - Enviar un email con errores a los operadores del Servicio de Impresión. - Guardar información de recuperación en el log del Monitor de impresión. 		

Tabla 11. Requisito RF-5

Requisito RF-6		Nombre: Crear Servicio Monitor Impresión	
Prioridad	FASE 1 Opcional	FASE 2 Requerido	FASE 3 Requerido
Descripción	El Monitor de Impresión debe funcionar como un Servicio Windows.		

Tabla 12. Requisito RF-6

Requisito RF-7		Nombre: Conservar ficheros de Spool	
Prioridad	FASE 1 Requerido	FASE 2 Requerido	FASE 3 Requerido
Descripción	Los servidores donde se instala el Monitor de Impresión deben tener configurada la cola de impresión para que se conserven los documentos después de su impresión.		

Tabla 13. Requisito RF-7

Requisito RF-8		Nombre: Envío log por email	
Prioridad	FASE 1 Opcional	FASE 2 Deseable	FASE 3 Requerido
Descripción	Envío de email informativo cuando se capturan detalles de un trabajo de impresión. Se debe poder desactivar el envío de emails.		

Tabla 14. Requisito RF-8

4.1.2.2 Requisitos no funcionales

Requisito RNF-1		Nombre: Acceso a red Servidores de Impresión	
Prioridad	FASE 1 Opcional	FASE 2 Requerido	FASE 3 Requerido
Descripción	Los servidores con dispositivos de impresión que se quieren monitorizar deben estar conectados a la red y deben tener acceso al servicio que implementa la capa CAD.		

Tabla 15. Requisito RNF-1

Requisito RNF-2		Nombre: Sistema Operativo Monitor Impresión	
Prioridad	FASE 1 Requerido	FASE 2 Requerido	FASE 3 Requerido
Descripción	Los servidores con dispositivos de impresión que se quieran monitorizar deben tener alguno de estos S.O: Windows Vista, 7, 8, 10, 2008 server o 2012 server.		

Tabla 16. Requisito RNF-2

Requisito RNF-3		Nombre: Usuario Servicio Monitor Impresión	
Prioridad	FASE 1 Opcional	FASE 2 Requerido	FASE 3 Requerido
Descripción	El servicio Monitor de impresión se tiene que ejecutar con un usuario Administrador (debe poder modificar propiedades de las colas impresión, interactuar con el Sistema de gestión de Eventos del sistema y con el Sistema de Archivos de Windows).		

Tabla 17. Requisito RNF-3

Requisito RNF-4		Nombre: GUI de desarrollo	
Prioridad	FASE 1 Requerido	FASE 2 Requerido	FASE 3 Requerido
Descripción	El servicio Monitor Impresión será desarrollado Con Visual Studio 15.		

Tabla 18. Requisito RNF-4

Requisito RNF-5		Nombre: Lenguaje programación	
Prioridad	FASE 1 Requerido	FASE 2 Requerido	FASE 3 Requerido
Descripción	Se utilizará C# como lenguaje de programación.		

Tabla 19. Requisito RNF-5

Requisito RNF-6		Nombre: Capa CAD acceso a BD Oracle	
Prioridad	FASE 1 Opcional	FASE 2 Requerido	FASE 3 Requerido
Descripción	<p>Es necesario un servidor con S.O Windows que admita .NET 4.x (Windows Vista, 7, 8, 10, 2008 Server o 2012 Server) y tenerlo instalado.</p> <p>El servidor debe contar además con:</p> <ul style="list-style-type: none"> - Un servidor de aplicaciones IIS 7.x o superior donde se instalará el Servicio que implementa la capa CAD. - Debe tener instalado un cliente Oracle con la configuración de acceso a la BD Oracle donde se almacenan los detalles de los trabajos de impresión. <p>Se utilizará C# para crear el servicio WCF que implementa la capa CAD utilizando la librería ODP.NET de Oracle (más detalles sobre ODP.NET en el apartado dedicado a ODP.NET en el anexo).</p> <p>En cuanto a configuración de red:</p> <ul style="list-style-type: none"> - El Servicio de la capa CAD debe ser accesible desde los equipos que tienen instalado el Servicio Monitor de impresión. - La BD Oracle donde se almacenan los detalles de los trabajos de impresión debe ser accesible desde el Servidor donde está instalado el Servicio de la capa CAD. 		

Tabla 20. Requisito RNF-6

Requisito RNF-7		Nombre: Herramienta soporte SCRUM	
Prioridad	FASE 1 Opcional	FASE 2 Opcional	FASE 3 Requerido
Descripción	Para la Fase3 se utilizará Visual Studio TFS como soporte para SCRUM.		

Tabla 21. Requisito RNF-7

Requisito RNF-8		Nombre: Habilar Eventos Sistema Impresión	
Prioridad	FASE 1 Requerido	FASE 2 Requerido	FASE 3 Requerido
Descripción	En los servidores donde está instalado el Monitor de Impresión tienen que estar habilitados los eventos del Sistema de Impresión a nivel Operativo a través de la herramienta del Visor de Eventos.		

Tabla 22. Requisito RNF-8

4.1.3 Casos de Uso

Los diagramas de casos de uso (Pérez, Ginestá, Matías, & Hernández, 2007, p. 84) nos permiten modelar gráficamente las relaciones de los distintos usos del sistema y sus participantes o actores.

Un diagrama de casos de uso consta de los siguientes elementos:

Actores: Representan los roles que juegan los usuarios y también otros sistemas que interactúan con nuestro sistema. En nuestro caso de estudio nuestro sistema interactúa principalmente con otros sistemas.



Ilustración 18. Diagrama caso de uso: Actor.

Caso de uso: Las acciones que puede realizar nuestro sistema. Para identificarlos es útil identificar las tareas y responsabilidades de cada actor.



Ilustración 19. Diagrama caso de uso: Caso de uso.

Las relaciones indican la actividad o flujo de información. Estas son las relaciones utilizadas:

Relación de asociación: Entre actores y casos de uso. Se representa con una flecha e indican si proporcionan o reciben información (según la dirección de la flecha).

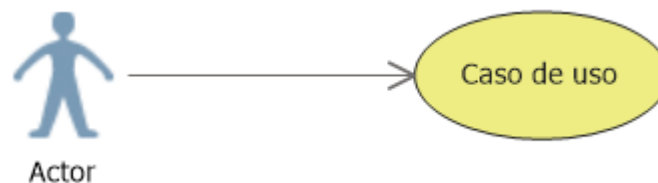


Ilustración 20. Diagrama caso de uso: Relación de asociación.

Relación de dependencia: La definición de un elemento depende de la definición de otro. Se representa con una flecha con línea discontinua.

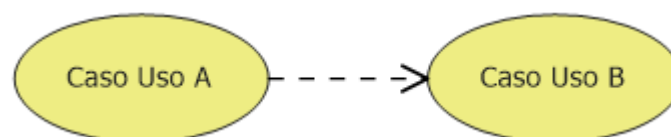


Ilustración 21. Diagrama caso de uso: Relación de dependencia.

Relación incluye: Un caso de uso invoca otro caso de uso. Se representa con una flecha con el texto <<incluye>>.

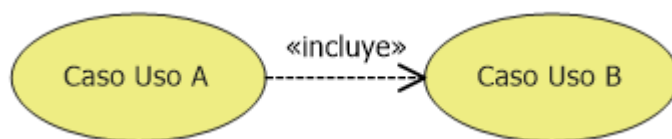


Ilustración 22. Diagrama caso de uso: Relación incluye.

Relación Extiende: un caso de uso amplía la definición de otro caso de uso en condiciones específicas. Se representa con una flecha con el texto <<extiende>>.

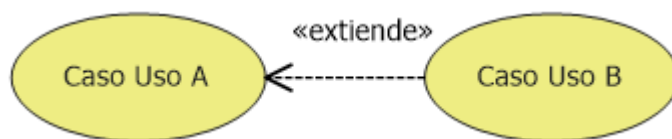


Ilustración 23. Diagrama caso de uso: Relación extiende.

Para describir los casos de uso se va a utilizar la siguiente ficha:

Identificador	<Identificador>
Nombre	<Nombre>
Actor	<Actor>
Objetivo	<Objetivo>
Precondiciones	<Precondiciones>
Fase	<Fase>
Escenario	<Escenario>

Tabla 23. Ficha para describir casos de uso.

Información contenida en la ficha:

- **Identificador** único para cada caso de uso.
- **Nombre** para cada caso de uso como aparece en el diagrama.
- **Actor** que realiza la acción, como aparece en el diagrama.
- **Objetivo** o función que realiza el caso de uso.
- **Precondiciones** o condiciones previas para poder realizar la acción.
- **Fase** en la que se implementa.
- **Escenario** o especificación de los pasos a seguir para realizar la acción.

4.1.3.1 Diagrama de casos de uso.

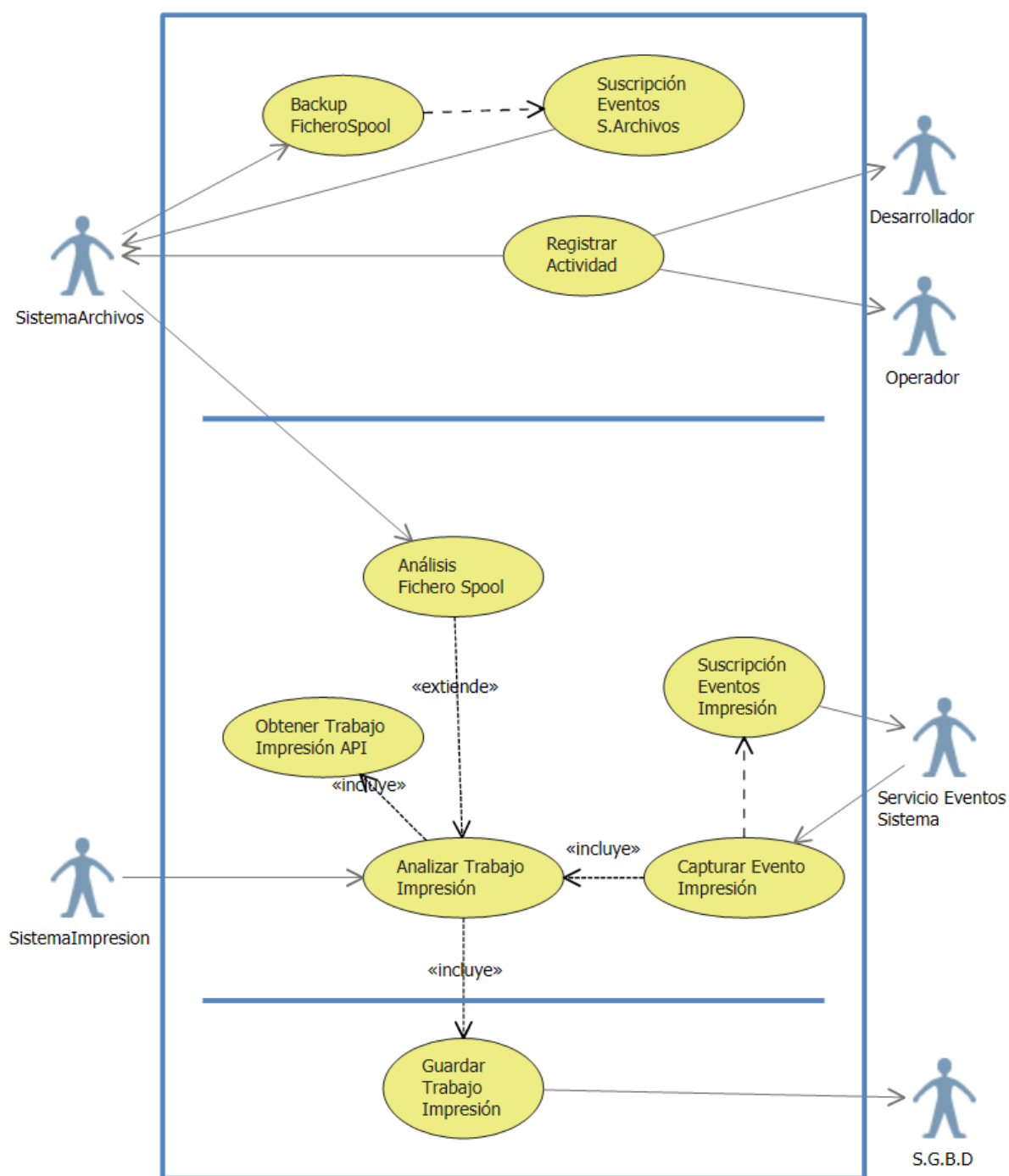


Ilustración 24. Diagrama casos de uso del Servicio Monitor Impresión.

No se ha incluido la relación <<incluye>> de todos los casos de uso sobre el caso de uso Registrar Actividad para no saturar el diagrama.

4.1.3.2 Especificación de casos de uso

Identificador	CU1
Nombre	Capturar Evento Impresión
Actor	Servicio Eventos Sistema

Objetivo	Detectar cuando se imprime un documento
Precondiciones	Estar suscrito a los eventos de impresión del registro de eventos de Windows.
Fase	Fase1
Escenario	Cuando se recibe un evento de impresión que indica que se ha imprimido un documento se obtiene el número de trabajo de impresión y la impresora.

Tabla 24. Especificación caso de uso. CU1

Identificador	CU2
Nombre	Suscripción Eventos impresión
Actor	Servicio Eventos Sistema
Objetivo	Capturar eventos de impresión
Precondiciones	El Servicio Monitor de Impresión se inicia con el sistema.
Fase	Fase1
Escenario	Cada vez que se inicia el Servicio Monitor de Impresión debe suscribirse a los eventos de impresión registrados por el Servicio de registro de eventos. El monitor de impresión debe seguir suscrito a los eventos de impresión mientras está activo.

Tabla 25. Especificación caso de uso. CU2

Identificador	CU3
Nombre	Analizar Trabajo Impresión
Actor	SistemaImpresion
Objetivo	Obtener los detalles de los trabajos de impresión
Precondiciones	Para analizar un trabajo de impresión es necesario identificarlo mediante la impresora y el número de trabajo de impresión.
Fase	Fase2
Escenario	Se realiza el análisis del trabajo de impresión identificado por el número de trabajo de impresión y la impresora. A través del número de trabajo de impresión y el nombre de la impresora se utiliza la API del sistema de impresión para obtener los detalles de impresión. Si no se obtienen todos los detalles de impresión necesarios se debe analizar el fichero de spool donde se almacena el trabajo de impresión. Se envía notificación por email, a operador y desarrollador, y se guarda la información de los detalles de impresión obtenida por cada método.

Tabla 26. Especificación caso de uso. CU3

Identificador	CU4
Nombre	Obtener trabajo Impresión API
Actor	SistemaImpresion
Objetivo	Obtener los detalles del trabajo de impresión a través de la API del Sistema de impresión.

Precondiciones	Para poder identificar el trabajo de impresión es necesario el nombre de la impresora y el número de trabajo de impresión. La impresora debe estar configurada para conservar los documentos después de su impresión (ficheros de Spool).
Fase	Fase1
Escenario	Este módulo interactúa con el Sistema de Impresión, utilizando la Interface de las API del Sistema de Impresión. Obtiene los detalles del trabajo de impresión identificado por su número de trabajo y por el nombre de la impresora.

Tabla 27. Especificación caso de uso. CU4

Identificador	CU5
Nombre	Análisis Fichero Spool
Actor	SistemaArchivos
Objetivo	Obtener los detalles del trabajo de impresión analizando el fichero de spool donde se guarda el trabajo de impresión.
Precondiciones	Es necesario conocer el número de trabajo de impresión y la carpeta de la cola de impresión. La impresora debe estar configurada para conservar los documentos después de su impresión (ficheros de Spool).
Fase	Fase1
Escenario	Realiza el análisis del fichero de Spool. Como resultado se obtienen detalles del trabajo de impresión que no se han podido obtener a través de la API del Sistema de impresión.

Tabla 28. Especificación caso de uso. CU5

Identificador	CU6
Nombre	Guardar Trabajo Impresión
Actor	S.G.B.D
Objetivo	Guardar en BD los detalles obtenidos de los trabajos de impresión.
Precondiciones	Haber obtenido los detalles del trabajo de impresión.
Fase	Fase2
Escenario	Recibe una instrucción SQL para lanzar sobre la BD y guardar los detalles del trabajo de impresión. Guarda registro de los detalles de impresión obtenidos por cada uno de los métodos empleados: API, Spool.

Tabla 29. Especificación caso de uso. CU6

Identificador	CU7
Nombre	Registrar Actividad
Actor	SistemaArchivos, Desarrollador, Operador.
Objetivo	Guardar, en ficheros de log, registro de todas las operaciones realizadas por el Servicio monitor de impresión. También enviar notificaciones por email con información de los trabajos de impresión y con los errores que se produzcan.
Precondiciones	N/A
Fase	Fase1

Escenario	Este módulo debe estar accesible desde el resto de los módulos del sistema, de modo que cualquier actividad que se produce es recibida y registrada. Esta puede ser registrada en ficheros de log o notificaciones de email.
------------------	--

Tabla 30. Especificación caso de uso. CU7

Identificador	CU8
Nombre	Backup Fichero Spool
Actor	SistemaArchivos
Objetivo	Guardar copia con fecha de los ficheros de Spool para poder realizar un análisis a posteriori del fichero de spool para cualquier trabajo de impresión.
Precondiciones	Recibir notificación de la creación de un fichero de Spool. La impresora debe estar configurada para conservar los documentos después de su impresión (ficheros de Spool).
Fase	Fase3
Escenario	El sistema de impresión no guarda indefinidamente los ficheros de spool y periódicamente los va sobrescribiendo. Por lo que este módulo guarda una copia de los ficheros de spool con la fecha de creación del fichero.

Tabla 31. Especificación caso de uso. CU8

Identificador	CU9
Nombre	Suscripción Eventos Sistema Archivos
Actor	SistemaArchivos
Objetivo	Recibir eventos de creación de ficheros de spool.
Precondiciones	Indicar el directorio de spool de la cola de impresión.
Fase	Fase3
Escenario	Cada vez que se inicia el Servicio Monitor de Impresión debe suscribirse a los eventos del Sistema de Archivos, concretamente a los de creación de ficheros en el directorio donde se guardan los ficheros de Spool de la cola de Impresión. Debe mantenerse registrado mientras que el Monitor de impresión está activo.

Tabla 32. Especificación caso de uso. CU9

4.2 Diseño y codificación.

👁 Ver anexo con Referencia de archivos del proyecto con código fuente

4.2.1 Lógica del Servicio Monitor Impresión.

En el siguiente diagrama de flujo se muestra la lógica a seguir para la obtención de información a través de las distintas fuentes:

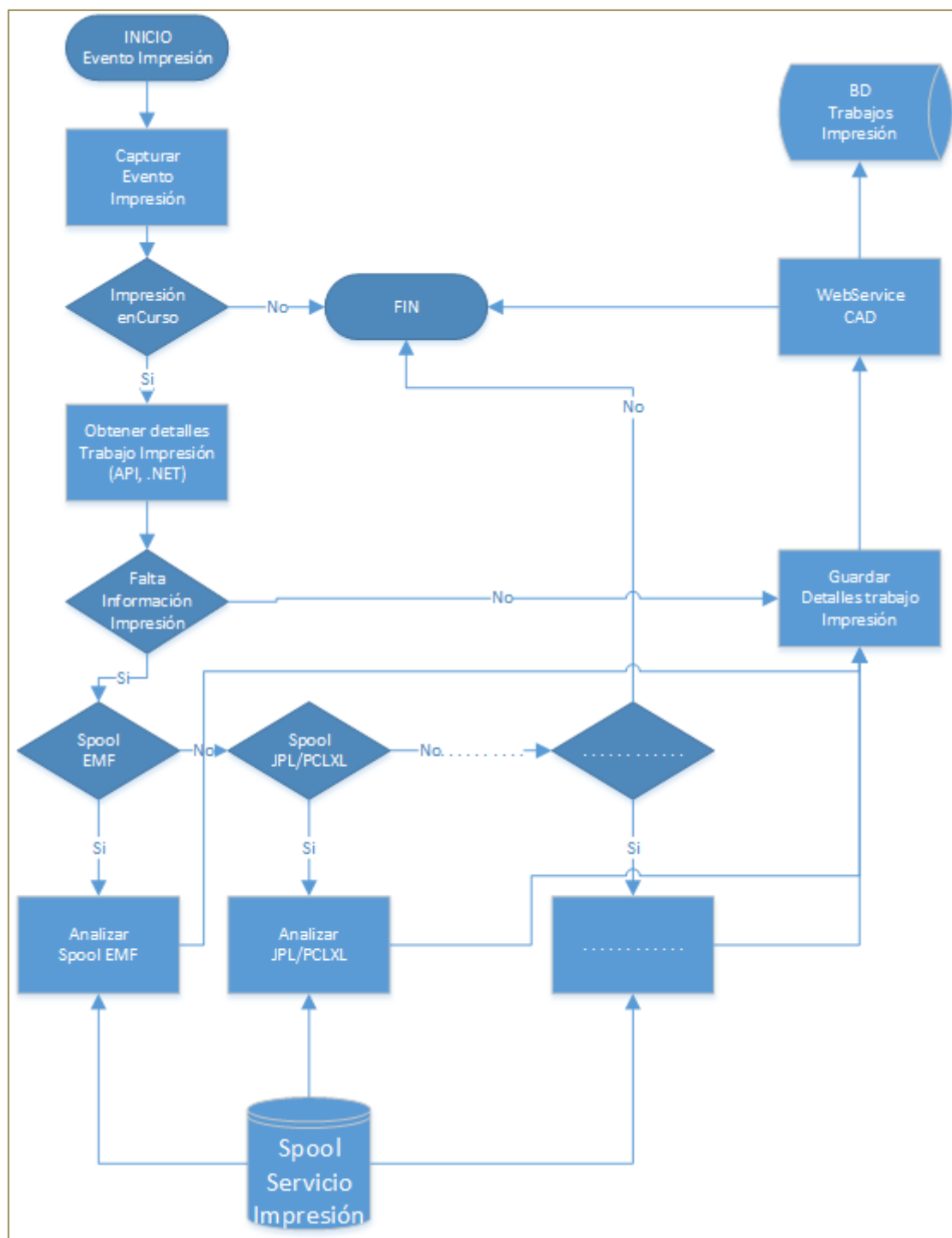


Ilustración 25. Flujograma. Captura propiedades de trabajo de impresión.

4.2.2 Diagrama de clases

En la siguiente ilustración se muestra la relación de las principales clases con las que se implementa el Servicio Monitor de Impresión. Se han incluido en el namespace MonitorTrabajosImpresion.

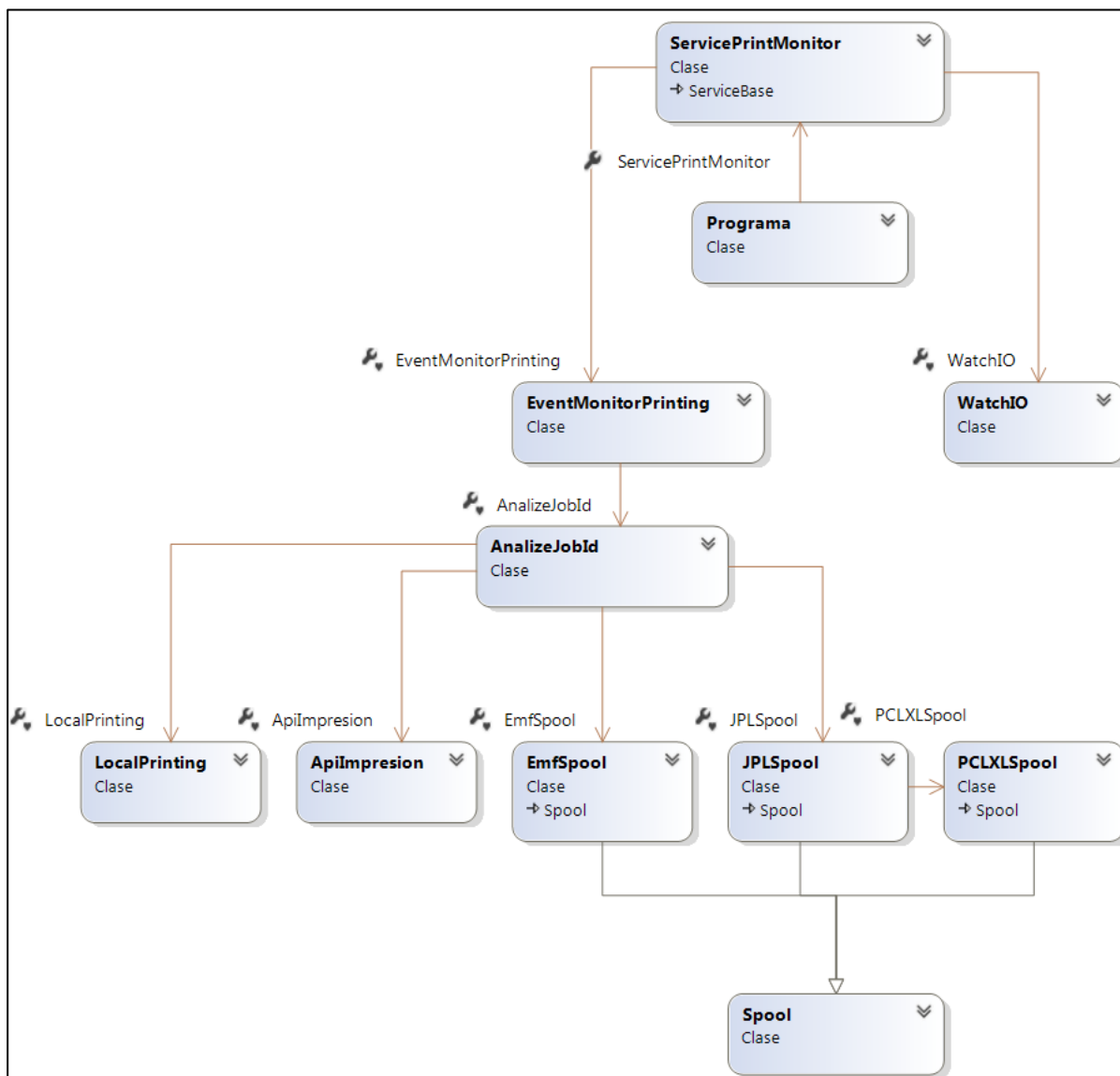


Ilustración 26. Diagrama de clases de Servicio Monitor Impresión

En la siguiente ilustración se muestra la relación de las principales clases con las que se implementa la capa de acceso a datos (servicio CAD). Se han incluido en el namespace WCFDB.



Ilustración 27. Diagrama de clases del Servicio WCF para la capa CAD.

En la siguiente tabla se muestra una breve descripción de cada una de las clases:

Clase	Descripción	Fase
Servicio Monitor de Impresión		
Programa	Clase que contiene el método Main() punto de inicio del programa.	Fase1
ServicePrintMonitor	Hereda de la clase ServiceBase de modo que esta clase se va a ejecutar como un servicio de Windows. Desde aquí se suscribe la clase EventMonitorPrinting para capturar eventos de Impresión y la clase WatchIO para capturar eventos de creación de ficheros de Spool.	Fase2
WatchIO	Esta clase se suscribe a los eventos del sistema de archivos. Su función es detectar la creación de ficheros de spool en la cola de impresión y hacer una copia de los ficheros de spool.	Fase2
EventMonitorPrinting	Clase para monitorizar los eventos del Sistema de Impresión. Con esta clase nos subscribimos a los eventos del Sistema de Impresión. Detectamos cuando se ha imprimido un documento y cual es su trabajo de impresión asociado. Esta información la pasamos a la clase adecuada con la que se extraen los detalles de impresión.	Fase1
AnalyzeJobId	Esta clase obtiene los detalles de los trabajos de impresión utilizando los métodos necarios para obtener dichos detalles. Los detalles de los trabajos de impresión se almacenan en BD utilizando el servicio WCF (implementado como Webservice) que implemente la capa CAD.	Fase2
LocalPrinting	Esta clase utiliza el Espacio de nombres System.Printing. Tras las pruebas realizadas se ha descartado para obtener detalles de trabajos de impresión. Solamente se utiliza para obtener un par de detalles: - El path de la cola de impresión - La fecha de impresión del trabajo de impresión indicado.	Fase1

Apilmpresion	Esta clase utiliza la interface de la API del Sistema de impresión para obtener los detalles de los trabajos de impresión. Además de configurar las impresoras para que conserven los ficheros de Spool después de su impresión.	Fase1
Spool	Clase para analizar ficheros de Spool	Fase3
EmfSpool	Hereda de la clase Spool e implemente el análisis de los ficheros de spool con contenido EMF.	Fase1
JPLSpool	Hereda de la clase Spool e implemente el análisis de los ficheros de spool con contenido JPL.	Fase1
PCLXLSpool	Hereda de la clase Spool e implemente el análisis de los ficheros de spool con contenido PCLXL.	Fase1

Servicio WCF (WebService) capa CAD

IDB	Interface del Servicio WCF (webservice) que implementa la capa CAD, donde se publican los métodos que serán accesibles desde el Servicio Monitor de Impresión.	Fase2
DB	Clase que implementa la interface IDB del servicio WCF. En concreto un par de métodos. El primero muy simple para probar el webservice; el segundo para lanzar instrucciones SQL a BD. Para interactuar con BD Oracle utiliza objetos de la clase DBOracle.	Fase2
DBOracle	Clase que implementa los métodos particulares para interactuar con BD Oracle.	Fase2

Tabla 33. Breve descripción de las clases más importantes.

En esta tabla resumen de clases, la columna Fase indica cuando se crea la clase aunque en posteriores fases se hayan añadido, eliminado o revisado sus funcionalidades.

Recordamos que las clases creadas en la Fase1 se crean como prototipos con la mínima funcionalidad para poder probar que es posible obtener los trabajos de impresión y sus propiedades. En la Fase2 se crean nuevas clases y se les da plena funcionalidad a los prototipos de la Fase1. En la Fase3 se identifican nuevos requisitos y se corrigen errores detectados.

Las clases tal como las presento en el título 4.3 (Implementación. Clases) es el resultado final tras la última fase.

4.2.3 BD

Para almacenar los trabajos de impresión en BD únicamente he empleado la tabla SPOOL_PRINTJOB con la siguiente estructura:

SPOOL_PRINTJOBS		
ID_JOBNAME	Varchar2(256)	NN (PK)
ID_FUENTE	Varchar2(64)	NN (PK)
N_JOB	Number(5,0)	NN
F_PRINTJOB	Date	
ID_LOGIN	Varchar2(30)	NN
ID_PRINTSERVER	Varchar2(256)	NN (PK)
ID_PRINTER	Varchar2(256)	NN (PK)
ID_DOCUMENT	Varchar2(1024)	NN
N_PAGES	Number(5,0)	NN
N_PAGES_PRINTED	Number(5,0)	
N_COLORPAGES	Number(5,0)	
N_LENGTH	Number(10,0)	NN
N_WIDTH	Number(10,0)	NN
ID_MEDIASIZE	Varchar2(256)	
N_MEDIASIZE	Number(5,0)	
ID_ORIENTATION	Varchar2(64)	
N_ORIENTATION	Number(1,0)	
ID_DUPLEX	Varchar2(30)	
N_DUPLEX	Number(1,0)	
ID_COLOR	Varchar2(30)	NN
N_COLOR	Number(1,0)	
N_COPIES	Number(3,0)	
ID_STATUS	Varchar2(256)	
ID_ISPLOTTER	Varchar2(1)	
ID_MEDIATYPE	Varchar2(256)	
N_MEDIATYPE	Number(5,0)	
N_JOBSIZE	Number(20,0)	
ID_NOTAS	Varchar2(1024)	
ID_REVISADO	Varchar2(1)	
F_PRINTJOB_USUARIO	Date	
ID_APENOM_USUARIO	Varchar2(256)	
ID_LOGIN_USUARIO	Varchar2(30)	
N_LENGTH_USUARIO	Number(5,0)	
N_WIDTH_USUARIO	Number(5,0)	
ID_MEDIASIZE_USUARIO	Varchar2(256)	
N_COPIES_USUARIO	Number(3,0)	
I_PRINTJOBS_ID_LOGIN (IX1)		
I_PRINTJOBS_N_JOB (IX2)		

Ilustración 28. Tabla Oracle SPOOL_PRINTJOBS para almacenar los trabajos de impresión

En el Servicio monitor de impresión no se hace ningún tratamiento de la información, de los trabajos de impresión, capturada por los distintos métodos (API, ficheros de Spool); de este modo se puede contrastar la información capturada a través de los distintos métodos y poder decidir para cada detalle de un trabajo de impresión el método que se va a emplear para obtenerlo.

Se deja para una etapa posterior, no objeto de este TFG, la explotación de la información capturada y por tanto el diseño de la BD donde guardar la información normalizada y sin redundancias almacenada en la tabla SPOOL_PRINTJOBS.

La tabla esta compuesta por los mismos campos de la estructura STRUCT_PRINT_JOB (descrita en el siguiente subtítulo). Pero además se han añadido un grupo de campos para guardar las anotaciones que los usuarios hacen a mano cuando imprimen un documento en los plotter del Servicio de Impresión del STIC, con el objeto de poder comparar la información que se captura

por el Servicio Monitor de Impresión y la información que anotan los usuarios a mano. Estos son los campos adicionales:

```
ID_NOTAS Varchar2(1024 ) // Comentarios al trabajo de impresión
ID_REVISADO Varchar2(1 ) DEFAULT 1 // Trabajo de impresión revisado in situ S/N
//Los siguientes campos recogen la información apuntada a mano por el usuario en las
hojas de registro de trabajos de impresión:
F_PRINTJOB_USUARIO Date //Fecha y hora de impresión del documento
ID_APENOM_USUARIO Varchar2(256 ) // Apellidos y nombre del usuario
ID_LOGIN_USUARIO Varchar2(30 ) //Cuenta de usuario
N_LENGTH_USUARIO Number(5,0) //Longitud de la hoja de impresión
N_WIDTH_USUARIO Number(5,0) //Ancho de la hoja de impresión
ID_MEDIASIZE_USUARIO Varchar2(256 ) // Tamaño de la hoja de impresión
N_COPIES_USUARIO Number(3,0) // Número de copias imprimidas del documento
```

4.2.4 Enumeraciones y estructuras empleadas para implementar las clases

Aunque, salvo algún detalle, en el Título dedicado a explicar las clases empleadas para codificar la solución no se muestra su codificación en C#, si se hace referencia a algunas estructuras utilizadas en las API del Sistema de Impresión Windows por eso parece conveniente mostrar aquí sus especificaciones en C# de la misma manera que se detallan las especificaciones de las funciones de las API del sistema de impresión utilizadas en C#.

Para una mejor comprensión del significado de los campos utilizados en las estructuras utilizadas en las APIs del sistema de impresión, consultar la referencia (MSDN Microsoft, n.d.-h).

Además se muestra la especificación de la estructura empleada para almacenar las propiedades de los trabajos de impresión adquiridas por los distintos métodos (API, análisis ficheros de Spool).

4.2.4.1 Estructuras

4.2.4.1.1 STRUCT_PRINT_JOB

Estructura empleada para guardar los detalles de los trabajos de impresión. En la propia estructura se da la explicación de cada campo.

```
public struct STRUCT_PRINT_JOB ///> Estructura con las propiedades de un trabajo de
impresión
{
    public string ID_JOBNAME;    // Código del trabajo de impresión. Se obtiene a
partir del Número de trabajo de impresión, y la fecha y hora del trabajo de impresión:
JJJJ(JobID)_YYYY(Año)/MM(Mes)/DD(Día)_HH(Hora):MiMi(Minuto):SS(Segundo)
    public string ID_FUENTE; // Método utilizado para obtener las propiedades de los
trabajos de impresión: System.Printing, API, JPL/PCLXL, EMF
    public int N_JOB; // Número de trabajo de impresión
    public string F_PRINTJOB; // Fecha de impresión
    public string ID_LOGIN; // Usuario
    public string ID_PRINTSERVER; // Nombre del Servidor de Impresión
    public string ID_PRINTER; // Nombre de impresora
    public string ID_DOCUMENT; // Nombre del documento a imprimir
    public int N_PAGES; // Número de páginas del documento
    public int N_PAGES_PRINTED; // Número de páginas impresas
    public int N_COLORPAGES; // Número de páginas impresas en color
    public int N_LENGTH; // Longitud de la página de impresión
```

```

public int N_WIDTH; // Ancho de la página de impresión
public string ID_MEDIASIZE; // Tamaño de la página
public int N_MEDIASIZE; // Código de tamaño de la página
public string ID_ORIENTATION; // Orientación de la página
public int N_ORIENTATION; // Código de orientación de la página
public string ID_DUPLEX; // Impresión simple o a doble cara
public int N_DUPLEX; // Código impresión simple o a doble cara
public string ID_COLOR; // Impresión a color o b/n
public int N_COLOR; // Código de impresión a color o b/n
public int N_COPIES; // Número de copias a imprimir del documento
public string ID_STATUS; // Resultado de la impresión
public string ID_ISPLOTTER; // Es un plotter S/N
public string ID_MEDIATYPE; // Tipo de papel
public int N_MEDIATYPE; // Código tipo de papel
public int N_JOBSIZE; // Tamaño en bytes del trabajo de impresión
} //PRINT_JOB

```

Detalle 27. Estructura STRUCT_PRINT_JOB para guardar los detalles de los trabajos de impresión.

4.2.4.1.2 JOB_INFO_2

Estructura utilizada con la función GetJob de la API del sistema de impresión para obtener los detalles del trabajo de impresión.

👁 *Se comentan los campos que contienen información relevante del trabajo de impresión. El resto no son de interés o no devuelven información fiable; p.e los campos TotalPages y PagesPrinted he comprobado analizando un número significativo de trabajos de impresión que no devuelven la información esperada.*

```

public struct JOB_INFO_2
{
    public uint JobId; // Número de trabajo de impresión
    public string pPrinterName; // Nombre de impresora
    public string pMachineName; // Nombre del servidor de impresión
    public string pUserName; // Usuario
    public string pDocument; // Nombre del documento
    public string pNotifyName;
    public string pDatatype;
    public string pPrintProcessor;
    public string pParameters;
    public string pDriverName; //Driver de impresión
    public IntPtr pDevMode; // Puntero a estructura DEVMODE. Varias de las propiedades
del trabajo de impresión se obtienen a partir de los campos contenidos en la
estructura DEVMODE
    public string pStatus; // Doc referencia no se corresponde ??
                                //public PSECURITY_DESCRIPTOR pSecurityDescriptor;
    public uint Status; // Resultado de la impresión
    public uint Priority;
    public uint Position;
    public uint StartTime;
    public uint UntilTime;
    public uint TotalPages;
    public uint Size; // Número de páginas
    public SYSTEMTIME Submitted;
    public uint Time;
    public uint PagesPrinted;

    public DEVMODE? DevMode
    {
        get
        {
            if (pDevMode != IntPtr.Zero)

```

```

        {
            return (DEVMODE)Marshal.PtrToStructure(pDevMode, typeof(DEVMODE));
        }
        else
        {
            return null;
        }
    }
    set
    {
        if (pDevMode != null)
        {
            Marshal.DestroyStructure(pDevMode, typeof(DEVMODE));
            Marshal.FreeHGlobal(pDevMode);
            pDevMode = IntPtr.Zero;
        }

        if (value != null)
        {
            pDevMode = Marshal.AllocHGlobal(Marshal.SizeOf(value));
            Marshal.StructureToPtr(value, pDevMode, false);
        }
    }
}
} // Fin JOB_INFO_2

```

Detalle 28. Estructura JOB_INFO_2. Utilizada con la API GetJob

4.2.4.1.3 SYSTEMTIME

Estructura conformada por los campos que componen una fecha: Año, Mes, Día de la semana, Día, hora, minuto y milisegundos.

```

public struct SYSTEMTIME
{
    public Int16 Year;
    public Int16 Month;
    public Int16 DayOfWeek;
    public Int16 Day;
    public Int16 Hour;
    public Int16 Minute;
    public Int16 Second;
    public Int16 Milliseconds;
}

```

Detalle 29. Estructura SYSTEMTIME. Para guardar una fecha y hora

4.2.4.1.4 PRINTER_INFO_5

Estructura utilizada con la función de la API SetPrinter para establecer/obtener la configuración de la impresora.

En concreto el campo Attributes se utiliza para consultar o establecer propiedades de configuración de impresora. Con el bit de la posición 9 establece si se conservan los documentos después de su impresión.

```

public struct PRINTER_INFO_5
{
    [MarshalAs(UnmanagedType.LPStr)]
    public string pPrinterName;
    [MarshalAs(UnmanagedType.LPStr)]
    public string pPortName;
    public uint Attributes;
}

```

```

    public uint device_not_selected_timeout;
    public uint transmission_retry_timeout;
}

```

Detalle 30. Estructura PRINTER_INFO_5

4.2.4.1.5 PRINTER_DEFAULTS

Lo utilizo con la API OpenPrinter para obtener el identificador interno de la impresora.

```

public struct PRINTER_DEFAULTS
{
    public string pDatatype;
    public IntPtr pDevMode;
    public PRINTER_ACCESS_MASK DesiredAccess;

    public DEVMODE? DevMode
    {
        get
        {
            if (pDevMode != IntPtr.Zero)
            {
                return (DEVMODE)Marshal.PtrToStructure(pDevMode, typeof(DEVMODE));
            }
            else
            {
                return null;
            }
        }
        set
        {
            if (pDevMode != null)
            {
                Marshal.DestroyStructure(pDevMode, typeof(DEVMODE));
                Marshal.FreeHGlobal(pDevMode);
                pDevMode = IntPtr.Zero;
            }

            if (value != null)
            {
                pDevMode = Marshal.AllocHGlobal(Marshal.SizeOf(value));
                Marshal.StructureToPtr(value, pDevMode, false);
            }
        }
    }
}

```

Detalle 31. Estructura PRINTER_DEFAULTS

4.2.4.1.6 DEVMODE

Descripción detallada de los campos de la estructura en la referencia (MSDN Microsoft, n.d.-b). Contiene información sobre la inicialización del dispositivo y entorno de una impresora. Esta estructura puede estar incluida dentro de otras estructuras. En concreto está incluida dentro JOB_INFO_2 y contiene la siguiente información relativa a un trabajo de impresión:

```

public struct DEVMODE
{
    [MarshalAs(UnmanagedType.ByValTStr, SizeConst = 32)]
    public string DeviceName;
    public ushort SpecVersion;
    public ushort DriverVersion;
    public ushort Size; // Número de páginas
    public ushort DriverExtra; //dmDriverExtra
}

```

```

    public uint Fields; // A nivel de bit significa cuales de los siguientes campos
han sido inicializados
    public short Orientation; // Orientación del papel
    public short PaperSize; // Tamaño del papel
    public short PaperLength; //Longitud del papel
    public short PaperWidth; // Anchura del papel
    public short Scale;
    public short Copies; // Número de copias del documento
    public short DefaultSource;
    public short PrintQuality;

    public short Color; // Impresión color o en b/n
    public short Duplex; // Impresión simple o a doble cara
    public short YResolution;
    public short TTOption;
    public short Collate;
    [MarshalAs(UnmanagedType.ByValTStr, SizeConst = 32)] //32 caracteres
    public string FormName;

    public ushort LogPixels; //reserved0
    public uint BitsPerPel; //reserved1
    public uint PelsWidth; //reserved2
    public uint PelsHeight; //reserved3

    public uint Nup;

    public uint DisplayFrequency; //reserved4

    public uint ICMMMethod;
    public uint ICMIntent;
    public uint MediaType; // Tipo de medio de impresión
    public uint DitherType;
    public uint Reserved1; //reserved5
    public uint Reserved2; //reserved6

    public uint PanningWidth; //reserved7
    public uint PanningHeight; //reserved8
}

```

Detalle 32. Estructura DEVMODE

4.2.4.2 Enumeraciones

4.2.4.2.1 EVENTS

Enumeración con los eventos de impresión y su significado.

```

public enum EVENTS
{
    PRINTER_PAUSED = 303, //Pausar cola impresión
    PRINTER_UNPAUSED = 304, //Reanudar cola impresión
    PRINTER_SET = 306, //Estableciendo configuracion de impresora
    DOCUMENT_PRINTED = 307, //Documento impreso
    DOCUMENT_RESUMED = 309, // Reanudando impresión de documento
    DOCUMENT_DELETED = 310, // Trabajo de impresión eliminado
    JOB_DIAG = 800, //Poniendo trabajo en cola
    JOB_DIAG_PRINTING = 801, //Imprimiendo
    DELETE_JOB_DIAG = 802, //Eliminando trabajo de impresión
    RENDER_JOB_DIAG = 805, //Presentando trabajo de impresión
    FILE_OP_FAILED = 812, //No se pueden eliminar ficheros de Spool
    PRINT_DRIVER_SANDBOX_JOB_PRINTPROC = 842, // El servidor de Impresión envió el
trabajo a la impresora
    _ID_EVEN_CONTROL = 9999, // Evento generado por la aplicación para chequear que se

```

```
están monitorizando los eventos  
}
```

Detalle 33. Enumeración EVENTS

4.3 Implementación. Clases

4.3.1 Clase EventMonitorPrinting

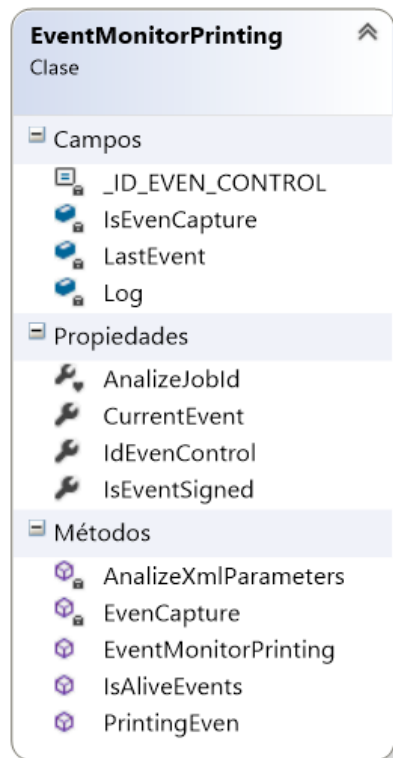


Ilustración 29. Clase EventMonitorPrinting.

Clase para monitorizar los eventos del Sistema de Impresión.

Con esta clase nos subscribimos a los eventos del Sistema de Impresión. Detectamos cuando se ha imprimido un documento y cual es su número de trabajo de impresión asociado. Esta información la pasamos a la clase adecuada con la que se extraen los detalles de impresión.

4.3.1.1 Documentación del constructor

Sin constructor.

4.3.1.2 Documentación de las funciones miembro

4.3.1.2.1 static void AnalyzeXmlParameters (Object eventoXML)[static], [private]

Análisis de los Eventos de Impresión.

Los eventos de impresión se reciben en formato XML. Por lo que se ha construido un método para analizar los eventos en formato XML. Para una mejor comprensión de XML se puede consultar (Reino Romero, 2011), (Harold & Means, 2005).

Parámetros:

<i>eventoXML</i>	objeto con el contenido XML del evento que se va a analizar
------------------	---

Comentarios:

Se definen los códigos de eventos de impresión en la enumeración: `ENUM_SYSTEM_PRINTING.EVENTS`

Si se ejecuta este método es por que está suscrito a los eventos del Sistema de Impresión Windows. Establecemos por tanto el atributo para controlar si se están capturando los eventos de impresión: `IsEvenCapture` como verdadero.

Se carga el contenido XML del evento en un objeto tipo documento XML (`XmlDocument`)

Obtenemos el evento.

Se analizan los siguientes eventos de impresión y se extraen los parámetros asociados:

- **Pausar cola impresion**
- **Reanudar Impresora pausada**
- **Estableciendo configuracion de Impresora:** Se comprueba que los trabajos de impresión se guardan en disco, sino, se vuelve a configurar la impresora para que se guarden (Con `ApiImpresion.KeepSpoolFiles(Printer)` aseguramos que conservan los documentos después de su impresión).
- **Se ha imprimido el documento:** Este evento en especial es muy importante por que cuando se detecta hacemos que se desencadene el proceso de análisis del trabajo de impresión con la creación de un objeto de la clase `AnalyzeJobId`. Este evento lleva asociadas las siguientes propiedades del Trabajo de impresión:
 - Número de trabajo de impresión
 - Nombre de documento
 - Usuario
 - Servidor de impresión
 - Impresora
 - Puerto
 - Tamaño en bytes del fichero del trabajo de impresión
 - Número de hojas
- **Reanudación impresión**
- **Trabajo de impresión eliminado**
- **Poniendo el trabajo en cola**
- **Imprimiendo trabajo**
- **Se elimina el trabajo de impresión**
- **Presentando trabajo de impresión**
- **No se pudo eliminar el archivo de spool del trabajo de impresión**
- **El Servidor de impresión envió el trabajo a la impresora**
- **Evento autogenerado por nuestra aplicación:** `_ID_EVENT_CONTROL = 9999`, para el control de subscripción a los eventos de Impresión.

4.3.1.2.2 `static void EvenCapture (object obj, EventRecordWrittenEventArgs arg)[static], [private]`

Método delegado con el que se capturan los eventos de impresión. Sigue la siguiente lógica:

- Se desencadena con un evento de impresión
- Guardamos en el log del servicio MonitorImpresión información del evento capturado.

- Se invoca el método "AnalyzeXmlParameters" al que se le pasa como parámetro la información del evento capturado en formato XML para su análisis.

4.3.1.2.3 static void IsAliveEvents (object sender, EventArgs Args)[static]

Con este método se comprueba si se están recibiendo los eventos de impresión. Si no se reciben, nos subscribimos a los eventos de impresión.

Sigue la siguiente lógica:

- Definimos un evento personalizado `_ID_EVEN_CONTROL = 9999`
- Se define el origen del evento: "MonitorPrinting".
- Se define el canal donde se encuentran los eventos de impresión: "Microsoft-Windows-PrintService/Operational", en el BD de Eventos del sistema.
- Se establece el texto informativo del evento creado: *"Control suscripción a los eventos de impresión desde la aplicación MonitorImpresión"*
- Registramos el evento de impresión personalizado.
- Chequeamos la variable de control: "IsEvenCapture" para ver si efectivamente estamos suscritos. Si estamos suscritos esta variable se tenía que haber actualizado a través del método suscrito `EvenCapture()` con el valor true. En caso de que no estemos suscritos lanzamos el método `PrintingEven()` para subscribir el método `IsEvenCapture()`.

4.3.1.2.4 static void PrintingEven ()[static]

Este método realiza la suscripción a los eventos de impresión.

Sigue la siguiente lógica:

- Se define el contexto o canal para Referenciar los eventos del Servicios Impresion: "Microsoft-Windows-PrintService/Operational".
- Se restringen los eventos que se quieren monitorizar, definidos en la enumeración `ENUM_SYSTEM_PRINTING.EVENTS`. Se construye una Query similar a esta: `string queryString = "[(System/EventID=303) or (System/EventID=304) or (System/EventID=306) or (System/EventID=307) or (System/EventID=310) or (System/EventID=800) or " + "(System/EventID=801) or (System/EventID=802) or (System/EventID=805) or (System/EventID=812) or (System/EventID=842) or (System/EventID=" + _ID_EVEN_CONTROL + ")]";`
- Se crea el objeto de consulta de eventos: "eventQueryImpresion" indicando el contexto o canal: "Channel" y la consulta con los eventos restringidos: "queryString".
- Creamos el objeto visor de eventos: "suscribeEventoImpresion" restringiéndolo a los eventos definidos en el objeto de consulta de eventos: "eventQueryImpresion".
- Suscribimos como delegado al objeto: "suscribeEventoImpresion" el método: "EvenCapture".
- Habilitamos la generación de los eventos para que lance el método delegado.

En el siguiente fragmento de C# se muestra el código para suscribirse a los eventos de impresión:

```
// Rama registro eventos de impresión
string Channel = "Microsoft-Windows-PrintService/Operational";

// Query con eventos a monitorizar
string queryString = "[(System/EventID=" +
```

```

((int)ENUM_SYSTEM_PRINTING.EVENTS.PRINTER_PAUSED).ToString() + "));
queryString += " or (System/EventID=" +
((int)ENUM_SYSTEM_PRINTING.EVENTS.PRINTER_UNPAUSED).ToString() + "));
queryString += " or (System/EventID=" +
((int)ENUM_SYSTEM_PRINTING.EVENTS.PRINTER_SET).ToString() + "));
queryString += " or (System/EventID=" +
((int)ENUM_SYSTEM_PRINTING.EVENTS.DOCUMENT_PRINTED).ToString() + "));
queryString += " or (System/EventID=" +
((int)ENUM_SYSTEM_PRINTING.EVENTS.DOCUMENT_RESUMED).ToString() + "));
queryString += " or (System/EventID=" +
((int)ENUM_SYSTEM_PRINTING.EVENTS.DOCUMENT_DELETED).ToString() + "));
queryString += " or (System/EventID=" +
((int)ENUM_SYSTEM_PRINTING.EVENTS.JOB_DIAG).ToString() + "));
queryString += " or (System/EventID=" +
((int)ENUM_SYSTEM_PRINTING.EVENTS.JOB_DIAG_PRINTING).ToString() + "));
queryString += " or (System/EventID=" +
((int)ENUM_SYSTEM_PRINTING.EVENTS.DELETE_JOB_DIAG).ToString() + "));
queryString += " or (System/EventID=" +
((int)ENUM_SYSTEM_PRINTING.EVENTS.RENDER_JOB_DIAG).ToString() + "));
queryString += " or (System/EventID=" +
((int)ENUM_SYSTEM_PRINTING.EVENTS.FILE_OP_FAILED).ToString() + "));
queryString += " or (System/EventID=" +
((int)ENUM_SYSTEM_PRINTING.EVENTS.PRINT_DRIVER_SANDBOX_JOB_PRINTPROC).ToString(
) + "));
queryString+=" or (System/EventID=" +
((int)ENUM_SYSTEM_PRINTING.EVENTS._ID_EVEN_CONTROL).ToString() + "]]";

// Objeto de consulta de eventos.
EventLogQuery eventQueryImpresion = new EventLogQuery(Channel,
System.Diagnostics.Eventing.Reader.PathType.LogName, queryString);

// Objeto visor de eventos restringiendo según query
EventLogWatcher suscribeEventoImpresion = new
EventLogWatcher(eventQueryImpresion);

// Se suscribe como delegado el método: "EvenCapture".
suscribeEventoImpresion.EventRecordWritten += new
EventHandler<EventRecordWrittenEventArgs>(EvenCapture);

// Habilitamos la generación de eventos para que lance el método delegado.
suscribeEventoImpresion.Enabled = true;

```

Tabla 34. Ejemplo código C# suscripción a Eventos de Impresión

4.3.1.3 Documentación de los datos miembro

Definimos el evento de impresión `_ID_EVEN_CONTROL` para controlar si estamos suscritos a los eventos de impresión.

const int _ID_EVEN_CONTROL = 9999[private]

Atributo para chequear que se capturan eventos de impresión.

bool IsEvenCapture = false[static], [private]

Atributo donde se guarda el último evento capturado.

int LastEvent [static], [private]

Logger para generación de logs: `EventMonitorPrinting`.

readonly ILog Log = LogManager.GetLogger("EventMonitorPrinting")[static], [private]

Propiedad que devuelve el último evento de impresión capturado.

int CurrentEvent[static], [get]

Contiene el valor del evento de control `_ID_EVEN_CONTROL`. Esta propiedad se utiliza en los test automatizados.

int IdEvenControl[static], [get]

Indica si se está suscrito a los eventos de impresión.

bool IsEventSigned[static], [get]

4.3.2 Clase AnalyzeJobId

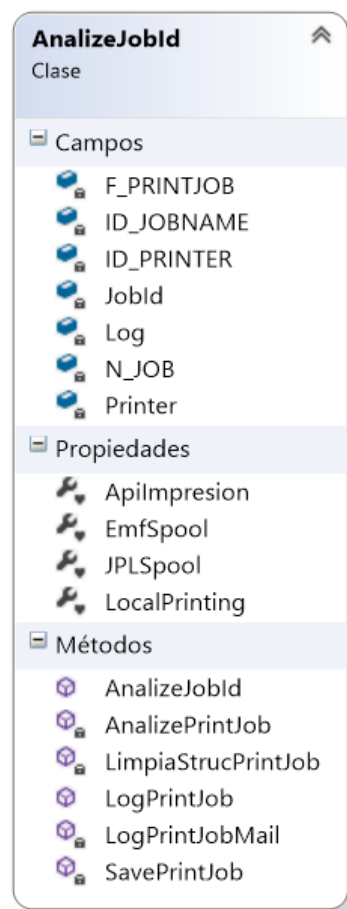


Ilustración 30. Clase `AnalyzeJobId`

Con esta clase se obtienen las propiedades de los trabajos de impresión.

Para ello se utilizan los distintos métodos implementados y se guarda para cada uno de ellos la información obtenida en BD.

4.3.2.1 Documentación del constructor

`AnalyzeJobId (string Printer, string JobId)`

Parámetros:

<i>Printer</i>	Nombre de la impresora por la que se ha imprimido el documento. Se guarda en el atributo de clase <i>JobId</i> .
<i>JobId</i>	Número de trabajo de impresión. Se guarda en el atributo de clase <i>Printer</i> .

Se sigue la siguiente lógica:

- Se guardan los parámetros de entrada en los atributos de clase homónimos.
- Se lanza el método *AnalyzePrintJob* para analizar el trabajo de impresión.

4.3.2.2 Documentación de las funciones miembro

4.3.2.2.1 void AnalizePrintJob () [private]

Implementa el análisis del trabajo de impresión guardando en B.D sus propiedades obtenidas a través de las API del Sistema de Impresión, del namespace System.Printing y del análisis de los ficheros de Spool. Sigue la siguiente lógica:

Mediante el nombre de impresora y el número de trabajo de impresión se obtienen las propiedades del trabajo de impresión utilizando el namespace System.Printing y:

- Se guarda registro en el Log del Servicio Monitor de Impresión.
- Se guarda la información obtenida en BD.

También mediante el nombre de impresora y el número de trabajo de impresión se obtienen las propiedades del trabajo de impresión a través de las APIs del Sistema de impresión y:

- Se guarda registro en el Log del Servicio Monitor de Impresión.
- Se guarda la información obtenida en BD.
- Se envía la información obtenida via email.

A continuación se procede al análisis del fichero de spool.

👁 El path del fichero de spool se obtiene mediante el método *LocalPrinting.PathPrintSpool()* y el nombre del fichero de Spool esta compuesto por el número del trabajo de impresión alineado con ceros a la izquierda en 5 posiciones con extensión *.SPL*. P.e. el nombre de fichero de spool del trabajo de impresión 23 sería *00023.SPL*.

Antes de realizar el análisis del fichero de Spool nos aseguramos de que el fichero no está bloqueado.

Si el fichero de spool contiene el trabajo de impresión en lenguaje EMF se realiza análisis basado en lenguaje EMFSPOOL y:

- Se guarda registro en el Log del Servicio Monitor de Impresión.
- Se guarda la información obtenida en BD.
- Se envía la información obtenida via email.

Si el fichero de spool contiene el trabajo de impresión en lenguaje JPL (se complementa con el análisis de un lenguaje PDL como HPGL2 o PCLXL como se ve en la clase *JPLSpool*) se realiza análisis basado en lenguaje JPL y:

- Se guarda registro en el Log del Servicio Monitor de Impresión.
- Se guarda la información obtenida en BD.
- Se envía la información obtenida via email.

En la siguiente ilustración se muestra el gráfico de llamadas desencadenado por el método:

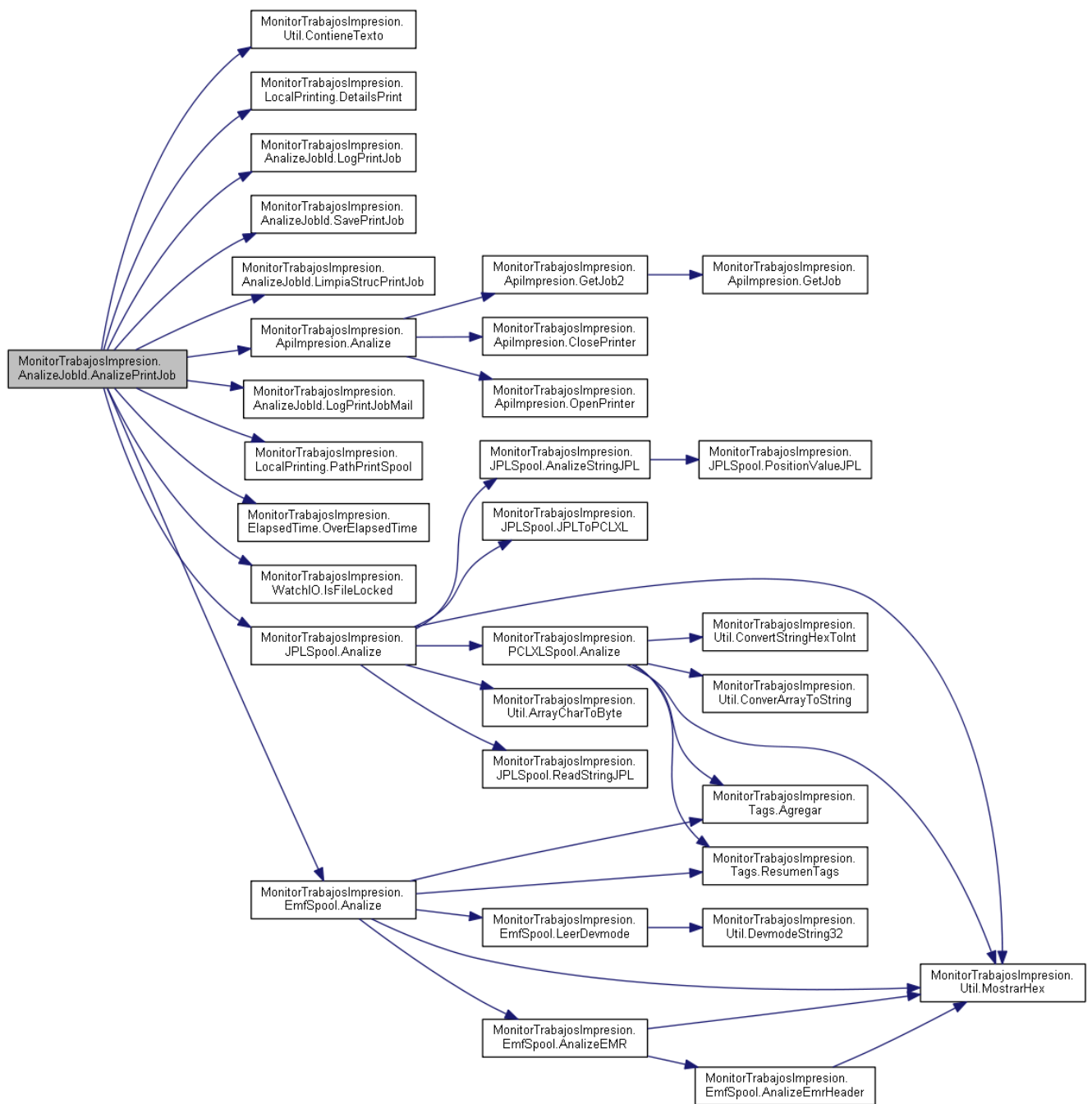


Ilustración 31. Gráfico de llamadas desencadenadas por el método AnalyzePrintJob.

4.3.2.2.2 void LimpiaStrucPrintJob (ref STRUCT_PRINT_JOB PrintJob) [private]

Inicializa las propiedades del Trabajo de Impresión de la estructura tipo STRUCT_PRINT_JOB pasada como parámetro por referencia.

Parámetros:

<i>PrintJob</i>	Estructura de tipo <code>STRUCT_PRINT_JOB</code> con las propiedades del trabajo de impresión.
-----------------	--

4.3.2.2.3 `static void AnalizeJobId.LogPrintJob (ref STRUCT_PRINT_JOB PrintJob) [static]`

Guarda en el log las propiedades del trabajo de impresión almacenadas en la estructura de tipo `STRUCT_PRINT_JOB` que se recibe como parámetro por referencia.

Parámetros:

<i>PrintJob</i>	Estructura de tipo <code>STRUCT_PRINT_JOB</code> con las propiedades del trabajo de impresión.
-----------------	--

4.3.2.2.4 `void LogPrintJobMail (ref STRUCT_PRINT_JOB PrintJob) [private]`

Envía un email con la información del trabajo de impresión, esta información se recibe en una estructura del tipo `STRUCT_PRINT_JOB`. Para ello se utiliza el logger (log4net) definido para la clase `AnalizeJoibId`.

Parámetros:

<i>PrintJob</i>	Estructura de tipo <code>STRUCT_PRINT_JOB</code> con las propiedades del trabajo de impresión.
-----------------	--

4.3.2.2.5 `void SavePrintJob (ref STRUCT_PRINT_JOB PrintJob) [private]`

Guarda en BD las propiedades del trabajo de impresión, esta información se recibe en una estructura del tipo `STRUCT_PRINT_JOB`. Para ello se hace referencia al servicio DB que implementa la capa CAD a través de un servicio WCF. En el anexo se da más detalles de los servicios WCF.

Parámetros:

<i>PrintJob</i>	Estructura de tipo <code>STRUCT_PRINT_JOB</code> con las propiedades del trabajo de impresión.
-----------------	--

Se sigue la siguiente lógica:

- Se construye la instrucción SQL de inserción con la información del trabajo de impresión recibidas en el parámetro del método, recibido por referencia.
- Se abre conexión con el WebService de la capa CAD.
- Se lanza la instrucción SQL.
- Cerramos la conexión con el servicio WCF de la capa CAD.

4.3.2.3 Documentación de los datos miembro

Propiedades del trabajo de impresión independientes de la fuente de obtención de información:

- **string ID_JOBNAME [private]**
- **string N_JOB [private]**
- **string ID_PRINTER [private]**
- **string F_PRINTJOB [private]**

ID_JOBNAME (identificador único de los trabajos de impresión) y F_PRINTJOB se establecen utilizando la fecha de impresión del documento, obtenida mediante el namespace System.Printing.

Estos atributos se incluyen en la estructura STRUCT_PRINT_JOB.

Al inicializar el objeto de la clase AnalizeJobId se establece el nombre de la impresora y el número de trabajo de impresión a analizar:

- **string Printer [private]**
- **string JobId [private]**

Logger para generación de logs: AnalizeJobId.

readonly ILog Log = LogManager.GetLogger("AnalizeJobId") [static], [private]

4.3.3 Clase LocalPrinting

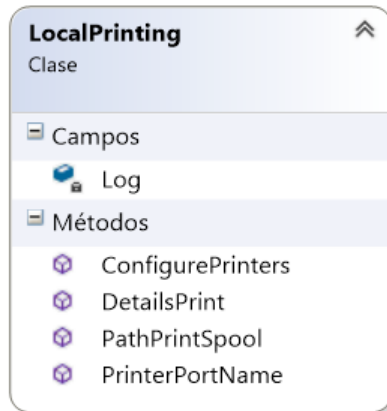


Ilustración 32. Clase LocalPrinting.

Con esta clase se obtienen propiedades del trabajo de impresión utilizando el namespace System.Printing.

El framework .NET 4.x incluye el namespace System.Printing (también incluido en versiones anteriores de .NET) que proporciona clases que permiten interactuar con el Sistema de impresión sin tener que utilizar directamente las API del Sistema de Impresión. Sin embargo no he podido obtener información fiable sobre los trabajos de impresión, además hay que tener en cuenta el apunte de Microsoft ya citado en el título 2 (MSDN Microsoft, n.d.-i) de los problemas que se puedan dar al utilizar este namespace en un servicio windows. No obstante lo sigo

incluyendo para algunas tareas para las que en la práctica no se ha producido ninguna incidencia.

4.3.3.1 Documentación del constructor

Sin constructor.

4.3.3.2 Documentación de las funciones miembro

4.3.3.2.1 static void ConfigurePrinters () [static]

Se establecen las propiedades de las impresoras gestionadas por el Servidor de impresión Local.

En concreto la propiedad "Conservar los documentos después de su impresión"; si no está activa se modifica para que lo esté. En la siguiente ilustración se puede ver la citada propiedad para la impresora virtual PDFCreator:

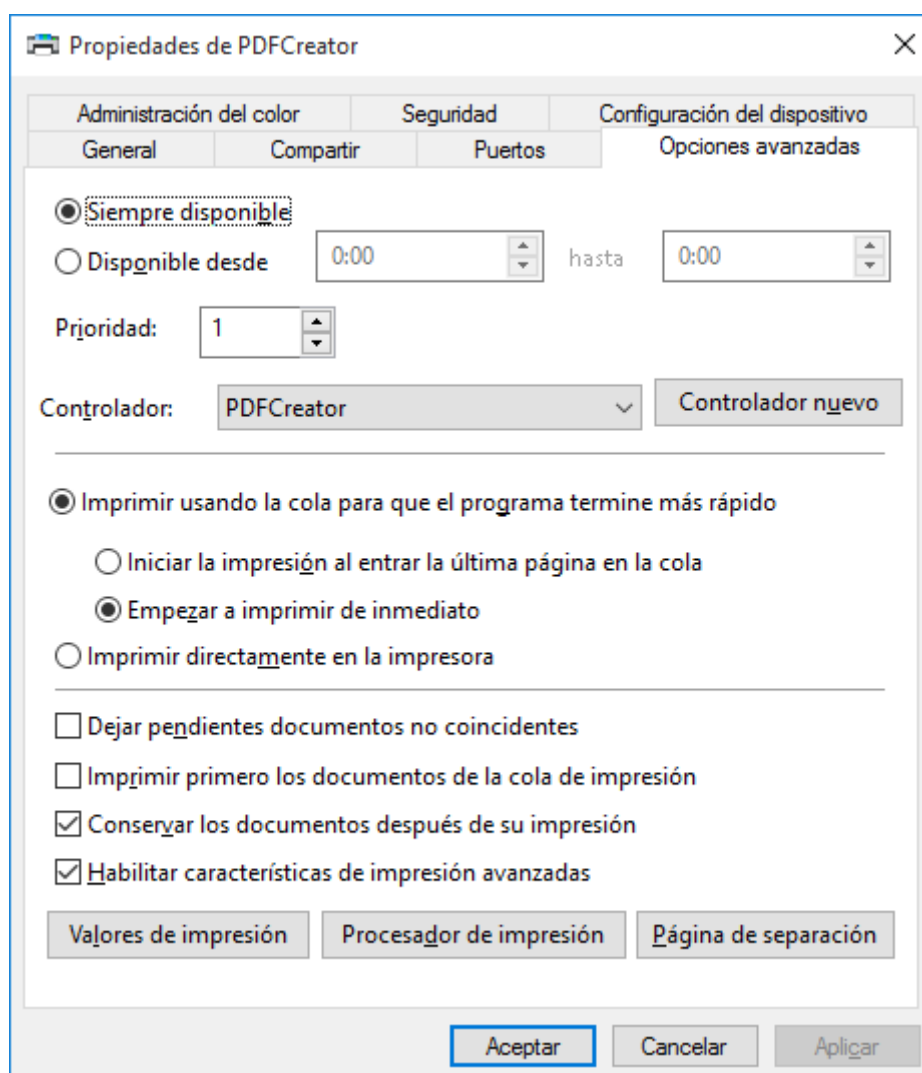


Ilustración 33. Propiedad de impresora para conservar documentos después de imprimir.

Se sigue la siguiente lógica:

Abrimos una conexión con el Servidor de Impresión Local.

Para cada impresora del Servidor de impresión se comprueba si tiene activa la propiedad para “Conservar los documentos después de su impresión” y si no lo está se activa. Para ello se crea un objeto `ApiImpresión` y se utiliza el método `KeepSpoolFiles()`.

```
4.3.3.2.2 static void DetailsPrint (string Printer, string JobId, ref STRUCT_PRINT_JOB PrintJob) [static]
```

Se obtienen detalles del Trabajo de Impresión a través del namespace `SystemPrinting`.

Parámetros:

<i>Printer</i>	Nombre de impresora
<i>JobId</i>	Número de trabajo de impresión
<i>PrintJob</i>	Estructura de tipo <code>STRUCT_PRINT_JOB</code> con las propiedades del trabajo de impresión.

Seguimos la siguiente lógica para obtener los detalles del trabajo de impresión identificado por el nombre de la impresora y el número de trabajo de impresión:

- Conectamos con el Servidor de impresión local.
- Abrimos la impresora según el parámetro de entrada (`Printer`).
- Obtenemos el trabajo de impresión lanzado por la impresora según el parámetro de entrada (`JobId`).
- Se guarda en el Log los detalles del Trabajo de Impresión.
- Guardamos los detalles del trabajo de impresión en la estructura (`PrintJob`) pasada por referencia.

```
4.3.3.2.3 static string PathPrintSpool () [static]
```

Es un método estático que devuelve como salida el Path del directorio utilizado por la cola de impresión del Servidor de Impresión Local.

```
4.3.3.2.4 static string PrinterPortName (string Printer) [static]
```

Es un método estático que devuelve el nombre del puerto de la impresora.

👁 *El puerto de la impresora se utiliza como parámetro en la función de la API del servicio de Impresión `SetPrinter` para establecer las propiedades de la impresora.*

Parámetros:

<i>Printer</i>	Nombre de la impresora de la que queremos obtener su puerto de impresión.
----------------	---

4.3.3.3 Documentación de los datos miembro

Logger para generación de logs: `AnalyzeJobId`.

`readonly ILog Log = LogManager.GetLogger("LocalPrinting") [static], [private]`

4.3.4 Clase Apilmpresion

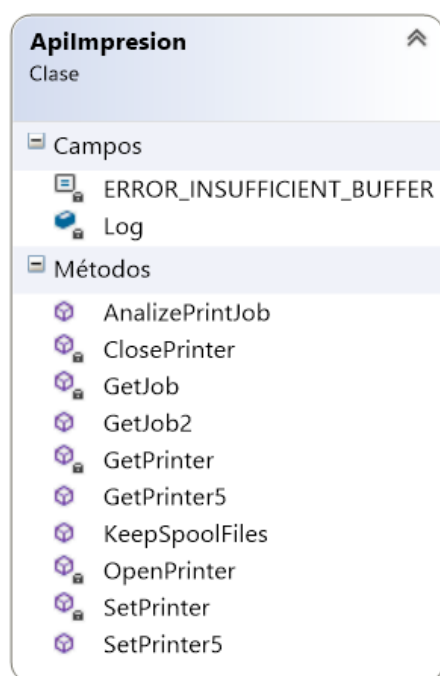


Ilustración 34. Clase Apilmpresión.

Con esta clase se obtienen los detalles de los trabajos de impresión utilizando funciones de la API del Servidor de Impresión. Estas funciones están disponibles a través de la librería `winspool.drv`.

4.3.4.1 Documentación del constructor

Sin constructor.

4.3.4.2 Documentación de las funciones miembro

4.3.4.2.1 `void Analyze (string Printer, uint JobId, ref STRUCT_PRINT_JOB PrintJob)`

Extrae las propiedades de un Trabajo de Impresión a través de las APIs proporcionadas a través de la librería `winspool.drv`.

Parámetros:

<i>Printer</i>	string con el nombre de la Impresora
<i>JobId</i>	Número del trabajo de impresión
<i>PrintJob</i>	struct tipo STRUCT_PRINT_JOB en el que se devuelven las propiedades del trabajo de impresión. Es una variable pasada por referencia.

Se sigue la siguiente lógica:

- Se abre la conexión con la impresora.
- Mediante el método `GetJob2` se recupera una estructura tipo `JOB_INFO_2` con las propiedades del trabajo de impresión.
- Se recuperan los valores del trabajo de impresión en el struct `STRUCT_PRINT_JOB`.
- Se cierra la conexión con la impresora.

Gráfico de llamadas para esta función:

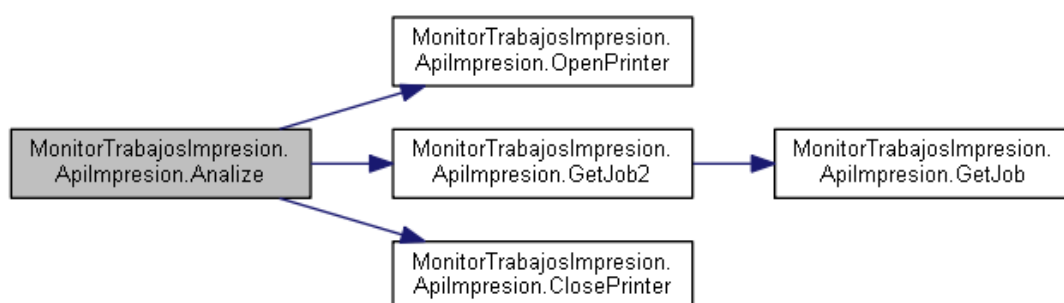


Ilustración 35. Gráfico de llamadas para el método `ApiPrintJob.Analyze`.

4.3.4.2.2 `static bool ClosePrinter (IntPtr hPrinter) [private]`

Función de la API del Servidor de impresión para cerrar la comunicación con una impresora. Se debe añadir el siguiente atributo para indicar la dll con las API del servidor de Impresión.

```
[DllImport("winspool.drv", SetLastError = true)] //Atributo de método
private static extern bool ClosePrinter(IntPtr hPrinter);
```

Detalle 34. API sistema impresión. `ClosePrinter`

Parámetros:

<i>hPrinter</i>	Identificador interno de impresora (obtenido a través de <code>OpenPrinter</code>)
-----------------	---

Comentarios:

Más detalles de la función en (MSDN Microsoft, n.d.-a)

4.3.4.2.3 `static bool GetJob (IntPtr hPrinter, uint JobId, uint Level, IntPtr pJob, uint cbBuf, ref uint pcbNeeded) [private]`

API para obtener las propiedades de un trabajo de impresión. Se debe añadir el siguiente atributo para indicar la dll con las API del servidor de Impresión.

```
[DllImport("winspool.drv", SetLastError = true)] //Atributo de método
private static extern bool GetJob(
    IntPtr hPrinter, // Lo obtenemos con la funcion OpenPrinter
    uint JobId, // Número de trabajo de impresión
    uint Level, // 1:JOB_INFO_1, 2:JOB_INFO_2
    IntPtr pJob, // Puntero a estructura indicada en Level
    uint cbBuf,
    ref uint pcbNeeded);
```

Detalle 35. API sistema impresión. `GetJob`

Parámetros:

<i>hPrinter</i>	Identificador interno de impresora (obtenido a través de OpenPrinter)
<i>JobId</i>	Número de trabajo de impresión
<i>Level</i>	1:JOB_INFO_1, 2:JOB_INFO_2. Según el detalle que queramos obtener del trabajo de impresión. Para nuestro trabajo necesitamos mayor detalles y utilizamos la de nivel 2.
<i>pJob</i>	Puntero a estructura indicada en Level
<i>cbBuf</i>	tamaño en bytes de la estructura apuntada por pJob
<i>pcbNeeded</i>	Puntero a una variable donde la función guarda el tamaño, en bytes, de los datos con información del trabajo de impresión.

Comentarios:

Si no conocemos el parámetro *cbBuf*, que indica el tamaño de la estructura utilizada para guardar información del trabajo de impresión según se indica en el parámetro *Level*, tenemos que utilizar la función de la siguiente forma:

```
// Si no conocemos el tamaño de la estructura apuntada por pJob: cbBuf, como parámetro
cbBuf lo ponemos a 0 y pJob también sin contenido, de esta manera obtenemos el tamaño
de la estructura apuntada por pJob en el parámetro pcbNeeded, es decir, el valor que
tenemos que tener en la variable cbBuf
if (GetJob(hPrinter, JobId, 2, IntPtr.Zero, 0, ref pcbNeeded))
{
    // No debería ejecutarse nunca el contenido del if
    Log.Error("No se ha obtenido correctamente tamaño del trabajo de impresión");
}
// Ahora se puede obtener el trabajo con los parámetros adecuados
if (GetJob(hPrinter, JobId, 2, pJob, pcbNeeded, ref pcbNeeded))
```

Detalle 36.API sistema impresión. Uso GetJob

Más detalles de la función en (MSDN Microsoft, n.d.-d).

4.3.4.2.4 JOB_INFO_2 GetJob2 (IntPtr *hPrinter*, uint *JobId*) [private]

Devuelve las propiedades de un Trabajo de impresión en una estructura *JOB_INFO_2* (MSDN Microsoft, n.d.-f).

Parámetros:

<i>hPrinter</i>	Impresora por la que se ha lanzado el Trabajo de Impresión. Identificador interno obtenido con la función OpenPrinter.
<i>JobId</i>	Número de trabajo de impresión

Devuelve:

JOB_INFO_2: Estructura con las propiedades del trabajo de impresión.

Se sigue la siguiente lógica

- Se obtiene las propiedades del trabajo de impresión en una estructura JOB_INFO_2.
- Se genera el log con la información contenido en la estructura JOB_INFO_2.
- Se devuelve la estructura JOB_INFO_2.

Gráfico de llamadas para esta función:



```
4.3.4.2.5 static bool GetPrinter (IntPtr hPrinter, uint Level, IntPtr pPrinter, uint cbBuf, out uint  
pcbNeeded)[private]
```

API para obtener las propiedades de una impresora. Se debe añadir el siguiente atributo para indicar la dll con las API del servidor de Impresión.

```
[DllImport("winspool.Drv", EntryPoint = "GetPrinterA", SetLastError = true,  
CharSet = CharSet.Ansi, ExactSpelling = true,  
CallingConvention = CallingConvention.StdCall)] //Atributo de método  
private static extern bool GetPrinter(  
IntPtr hPrinter,  
uint Level,  
IntPtr pPrinter,  
uint cbBuf,  
out uint pcbNeeded);
```

Detalle 37. API sistema impresión. GetPrinter

Parámetros:

<i>hPrinter</i>	Identificador interno de impresora (obtenido a través de OpenPrinter)
<i>Level</i>	Nivel o tipo de estructura utilizada para obtener información de impresora. Entre 1-9 niveles.
<i>pPrinter</i>	Estructura de datos utilizada según el nivel que se indica en el parámetro Level. (Utilizamos la estructura:PRINTER_INFO_5
<i>cbBuf</i>	tamaño en bytes de la estructura apuntada por pPrinter
<i>pcbNeeded</i>	Puntero a una variable donde la función guarda el tamaño, en bytes, de los datos con información de la impresora.

Comentarios:

Más detalles en el MSDN de Microsoft (MSDN Microsoft, n.d.-e).

```
4.3.4.2.6 PRINTER_INFO_5 GetPrinter5 (IntPtr WphPrinter)[private]
```

Con esta función se obtiene la propiedad de la impresora que se indica en el parámetro de entrada. Para ello utiliza la función `GetPrinter()` de la API del Servidor de Impresión.

Parámetros:

<i>WphPrinter</i>	Nombre de la impresora
-------------------	------------------------

Devuelve:

Estructura tipo `PRINTER_INFO_5` con información de la impresora.

Gráfico de llamadas para esta función:



Ilustración 36. Gráfico de llamadas de la función `GetPrinter5`.

4.3.4.2.7 void KeepSpoolFiles (string *Printer*)

Basado en el artículo (umeshb, n.d.).

Comprueba la propiedad de la impresora que indica si se conservan los documentos después de su impresión (ficheros de spool). Si no se conservan se modifica esta propiedad para que se haga.

Parámetros:

<i>Printer</i>	Nombre de la impresora.
----------------	-------------------------

Para ello se sigue la siguiente lógica:

- Abrimos la impresora y obtenemos su identificador interno.
- Obtenemos las propiedades de la impresora en una estructura de tipo `PRINTER_INFO_5`.
- Cerramos la impresora.
- Si no se guardan los documentos después de su impresión (para ellos se chequean los atributos obtenidos en la estructura de tipo `PRINTER_INFO_5`):
 - Se modifica el atributo en la estructura de tipo `PRINTER_INFO_5` para que conserven los documentos después de su impresión.
 - Abrimos la impresora con permiso de administrador para modificar sus propiedades
 - Establecemos las propiedades.
 - Cerramos la impresora.

Gráfico de llamadas para esta función:

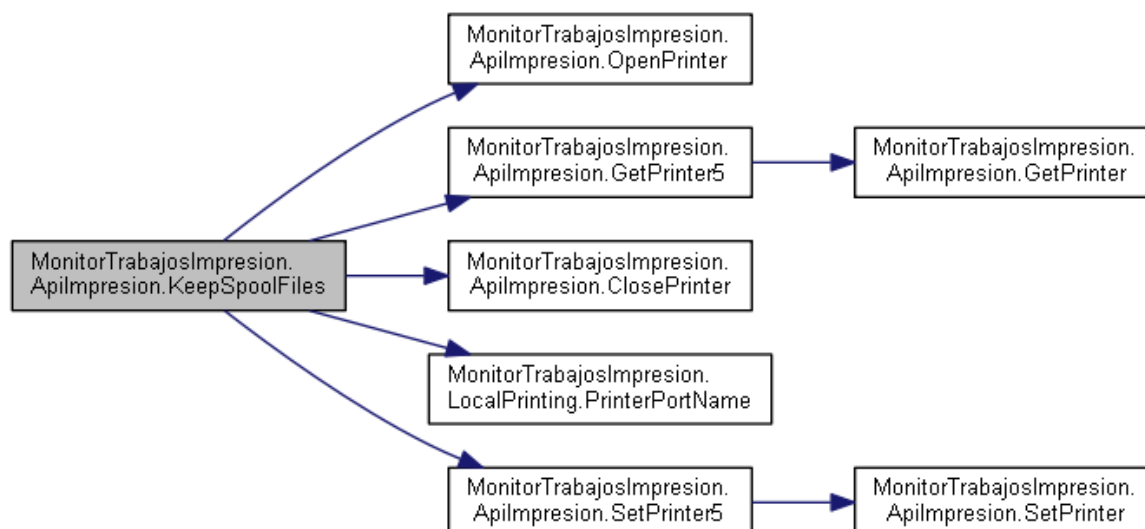


Ilustración 37. Gráfico de llamadas para la función KeepSpoolFiles.

4.3.4.2.8 static bool OpenPrinter (string pPrinterName, out IntPtr phPrinter, ref STRUCT_API_PRINTING.PRINTER_DEFAULTS pPrinterDefaults)[private]

API para poder interactuar con una impresora. Se debe añadir el siguiente atributo para indicar la dll con las API del servidor de Impresión.

```
[DllImport("winspool.drv", SetLastError = true)] //Atributo de método
private static extern bool OpenPrinter(
    string pPrinterName,
    out IntPtr phPrinter,
    ref STRUCT_API_PRINTING.PRINTER_DEFAULTS pPrinterDefaults);
```

Detalle 38. API sistema impresión. OpenPrinter

Parámetros:

<i>pPrinterName</i>	Nombre de la impresora
<i>phPrinter</i>	Identificador interno de impresora
<i>pPrinterDefaults</i>	Puntero a estructura PRINTER_DEFAULTS(características de la impresora)

Comentarios:

Más detalles sobre esta función en el MSDN de Microsoft (MSDN Microsoft, n.d.-g).

4.3.4.2.9 static bool SetPrinter (IntPtr hPrinter, uint Level, IntPtr pPrinter, uint Command)[private]

Función de la API del Servidor de Impresión para modificar las propiedades de una impresora. Se debe añadir el siguiente atributo para indicar la dll con las API del servidor de Impresión.

```
[DllImport("winspool.drv", SetLastError = true,
 CharSet = CharSet.Auto)] //Atributo de método
private static extern bool SetPrinter(
    IntPtr hPrinter,
    uint Level,
    IntPtr pPrinter,
    uint Command);
```

Detalle 39. API sistema impresión. SetPrinter

Parámetros:

<i>hPrinter</i>	Identificador interno de impresora (obtenido a través de OpenPrinter)
<i>Level</i>	Nivel o tipo de estructura utilizada para obtener información de impresora. Entre 1-9 niveles.
<i>pPrinter</i>	Estructura de datos utilizada según el nivel que se indica en el parámetro Level. (Utilizamos la estructura: PRINTER_INFO_5)
<i>Command</i>	Si el valor es 0. La impresora se reconfigura con los valores contenidos en la estructura apuntada por pPrinter

Comentarios:

Más detalle sobre esta función consultar (MSDN de Microsoft, n.d.)

4.3.4.2.10 void SetPrinter5 (IntPtr WphPrinter, ref IntPtr pPrinterInfo5) [private]

Establece las propiedades de la impresora.

Parámetros:

<i>WphPrinter</i>	Identificador interno de la impresora
<i>pPrinterInfo5</i>	Puntero a una estructura PRINTER_INFO_5.

Se sigue la siguiente lógica:

- Se almacena en una estructura de tipo PRINTER_INFO_5 la información referenciada por el puntero pPrinterInfo5.
- Se modifican las propiedades de la impresora con el contenido de la estructura de tipo PRINTER_INFO_5.
- Si no se pueden actualizar las propiedades de la impresora se genera una excepción y se guarda registro del error en el log.

4.3.4.3 Documentación de los datos miembro

Cuando se utilizan las funciones de la API del Servidor de impresión `GetPrinter` y `GetJob` sin especificar el tamaño del buffer que se quiere leer se produce el código de error definido en la siguiente constante:

```
const int ERROR_INSUFFICIENT_BUFFER = 122 [private]
```

Definición del logger para la clase `ApiImpresion`:

```
readonly ILog Log = LogManager.GetLogger("ApiImpresion") [static], [private]
```

4.3.5 Clase Spool

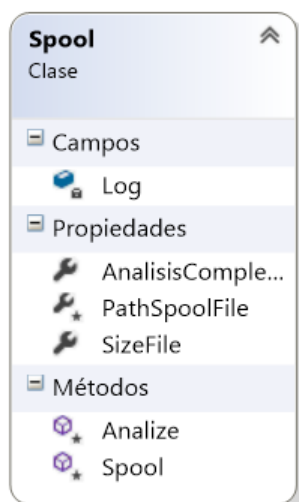


Ilustración 38. Clase Spool.

Clase base para análisis de ficheros de Spool (EmfSpool, JPLSpool y PCLXLSpool).

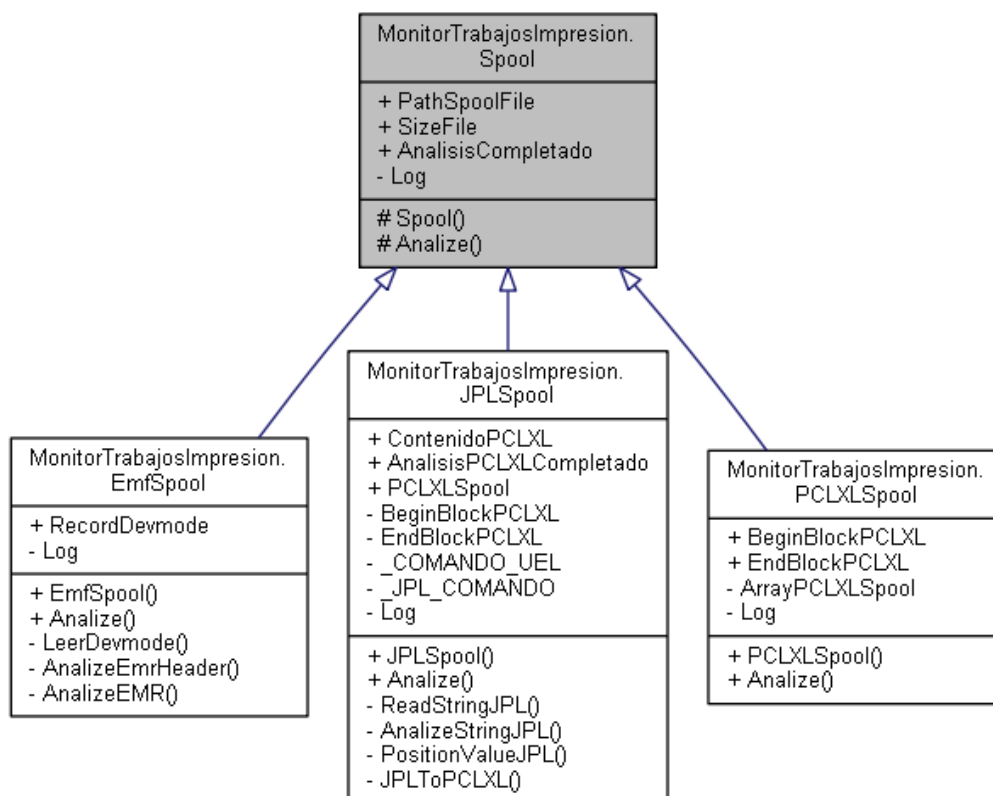


Ilustración 39. Diagrama de herencia sobre la clase Spool.

4.3.5.1 Documentación del constructor

Spool (string *PathFile*) [protected]

En el constructor se guarda el Path del fichero de spool y el tamaño.

Parámetros:

<i>PathFile</i>	Path del fichero de spool a analizar
-----------------	--------------------------------------

Se sigue la siguiente lógica:

- Se establece el directorio de la cola de impresión.
- Si existe el fichero de Spool se guarda su tamaño, necesario para realizar un análisis correcto del mismo.
- Si no existe el fichero de spool se guarda registro del error en el log y se genera una excepción con el Error.

4.3.5.2 Documentación de las funciones miembro

4.3.5.2.1 void Analyze (ref STRUCT_PRINT_JOB *PrintJob*) [protected]

Con este método se realiza el análisis del fichero de **Spool**.

Su implementación se desarrolla en las clases hijas.

4.3.5.3 Documentación de los datos miembro

Logger para generación de logs: Spool.

readonly ILog Log = LogManager.GetLogger("Spool") [static], [private]

Indica si se ha completado el análisis del fichero de Spool:

bool MonitorTrabajosImpresion.Spool.AnalisisCompletado [get], [set]

Path del fichero de spool a analizar:

string MonitorTrabajosImpresion.Spool.PathSpoolFile [get], [set], [protected]

Tamaño, en bytes, del fichero a analizar:

int MonitorTrabajosImpresion.Spool.SizeFile [get], [set]

4.3.6 Clase EmfSpool

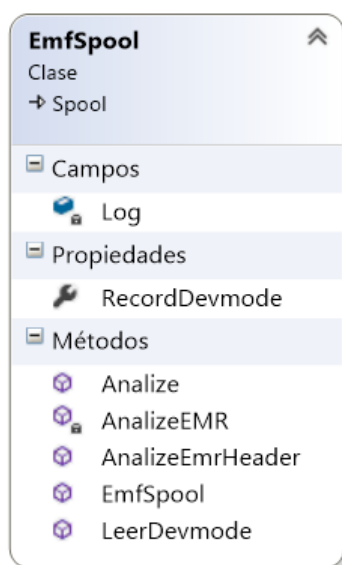


Ilustración 40. Clase EmfSpool.

Hereda de la clase Spool. Con esta clase se analizan los ficheros de spool con contenido EMF.

4.3.6.1 Documentación del constructor

EmfSpool (string PathFile)

Parámetros:

<i>PathFile</i>	ruta fichero de spool formato Emf
-----------------	-----------------------------------

4.3.6.2 Documentación de las funciones miembro

4.3.6.2.1 new void Analyze (ref STRUCT_PRINT_JOB *PrintJob*)

Se analiza el fichero de spool con contenido EMFSPOOL para extraer los detalles relevantes de impresión.

Como detalles de impresión importantes en este análisis se extrae:

- El número páginas impresas con contenido en color.
- El tamaño de página (para algunos plotter no se obtiene correctamente el tamaño a través de la API del servidor de impresión).

Parámetros:

<i>PrintJob</i>	Estructura pasada por referencia para guardar las propiedades del trabajo de impresión
-----------------	--

Se sigue la siguiente lógica:

- Apertura fichero de Spool.
- Análisis fichero de Spool.
 - Mientras existan registros EMFSPOOL:
 - Nos posicionamos para leer el siguiente registro EMFSPOOL.
 - Obtenemos el tipo y tamaño de registro EMFSPOOL.
 - Se procede al análisis del registro EMFSPOOL:
 - Si es registro EMRI_DEVMODE guardamos en el Log su contenido y guardamos propiedades del trabajo de impresión en la estructura con las propiedades del trabajo de impresión.
 - Si es un registro EMFSPOOL con contenido de página, realizamos un análisis EMF de su contenido. Para ello se utiliza el método *AnalyzeEMR()*.
 - Si el registro EMFSPOOL es del tipo EMRI_METAFILE_EXT -> El contenido de la Pagina es en color.
- Se cierra el fichero de Spool.
- Se genera log con el resumen del análisis EMF del fichero de Spool.

4.3.6.2.2 void AnalyzeEMR (ref BinaryReader *EmfFile*, int *positionMetafileRecord*, int *Size*, ref Tags *ResumenEMR*, ref STRUCT_PRINT_JOB *PrintJob*) [private]

Realiza el análisis EMF de un registro EMFSPOOL que almacena el contenido de una página (Metafile record).

Parámetros:

<i>EmfFile</i>	referencia al fichero de spool EMF
<i>positionMetafileRecord</i>	Posición comienzo Metafile
<i>Size</i>	tamaño del registro a analizar
<i>ResumenEMR</i>	Objeto de la clase Tags para guardar un resumen de los Tags

	utilizados
<i>PrintJob</i>	Estructura pasada por referencia para guardar las propiedades del trabajo de impresión

👁 *La clase Tags es una clase no principal cuya función es contabilizar el número de ocurrencias de cada tipo de elemento que se va analizando para poder mostrar un resumen del análisis. También se utilizan en el análisis PCLXL.*

Se sigue la siguiente lógica:

- Se recorre el registro EMFSPool con contenido de página, identificando los registros EMR de definición del contenido de página.
- Mientras existan registros EMF nos posicionamos en el siguiente y lo leemos.
 - Se genera log con información de los registros EMF identificados.
 - Si es un registro de cabecera lo analizamos con el método `AnalyzeEmrHeader()`.

4.3.6.2.3 `void AnalyzeEmrHeader (byte [] ArrayEmrHeader, ref STRUCT_PRINT_JOB PrintJob, int Position) [private]`

Lectura y análisis del registro de cabecera EMR_HEADER. Ver estructura en el título dedicado al estado del arte o en la siguiente referencia (“[MS-EMF]: Enhanced Metafile Format,” n.d., p. 115).

Parámetros:

<i>ArrayEmrHeader</i>	Array con el contenido del registro EMR_HEADER
<i>PrintJob</i>	Estructura pasada por referencia para guardar las propiedades del trabajo de impresión
<i>Position</i>	Posición del registro EMR_HEADER en el fichero de Spool

De su análisis se obtiene el tamaño de la página del documento. Más concretamente se obtienen las medidas del menor rectángulo que permite encerrar el contenido de la página.

4.3.6.2.4 `void LeerDevmode (ref BinaryReader EmfFile, int PositionDEVMO, ref STRUCT_PRINT_JOB PrintJob) [private]`

Lectura y análisis de registro DEVMODE, a partir de la posición dada como parámetro de entrada. Su estructura se define en el documento (“[MS-RPRN]: Print System Remote Protocol,” n.d., p. 33)

Parámetros:

<i>EmfFile</i>	referencia a fichero EMF
<i>PositionDEVMO</i> <i>DE</i>	posición del registro DEVMODE en el fichero EMF

<i>PrintJob</i>	Estructura pasada por referencia para guardar las propiedades del trabajo de impresión
-----------------	--

Se sigue la siguiente lógica:

- Se obtiene el contenido DEVMODE.
- Se guarda en el log el contenido del registro DEVMODE.
- Se actualizan las propiedades del trabajo de impresión.

4.3.6.3 Documentación de los datos miembro

Logger para generación de logs: EMFSpool

readonly ILog Log = LogManager.GetLogger("EmfSpool") [static], [private]

4.3.7 Clase JPLSpool

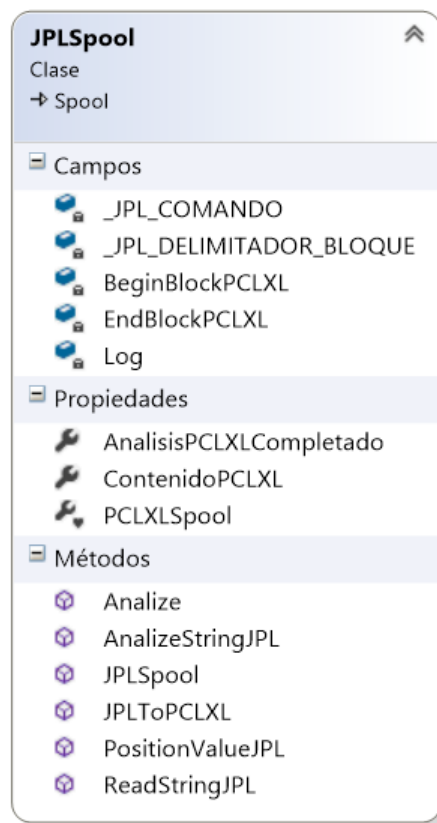


Ilustración 41. Clase JPLSpool.

Hereda de la clase Spool. Realiza el análisis de ficheros de spool con contenido JPL.

4.3.7.1 Documentación del constructor

JPLSpool (string PathFile)

En el constructor se guarda el Path del fichero de spool y el tamaño, heredado del objeto base.

Se comprueba si existe el fichero de spool de impresión y se guarda el Path.

Parámetros:

<i>PathFile</i>	Path del fichero de spool a analizar
-----------------	--------------------------------------

4.3.7.2 Documentación de las funciones miembro

4.3.7.2.1 new void Analyze (ref STRUCT_PRINT_JOB *PrintJob*)

Se analiza el fichero de spool en busca del comando UEL e instrucciones JPL para su análisis.

Parámetros:

<i>PrintJob</i>	Estructura del tipo STRUCT_PRINT_JOB pasada por referencia en la que se guardan los detalles del trabajo de impresión obtenidos.
-----------------	--

Se sigue la siguiente lógica:

Buscar cadenas del bloque JPL en el fichero de spool de impresión. Se localizan cadenas que comienzan por:

- "%-12345X@PJL": Comando UEL.
- "@PJL": Instrucciones JPL.

Se analiza el contenido de las instrucciones JPL.

Si se identifica que dentro del contenido JPL se conmuta a lenguaje PCLXL para definir el contenido del documento, se guarda la posición de inicio y fin del Bloque PCLXL en las variables:

- BeginBlockPCLXL
- EndBlockPCLXL

Si se utiliza lenguaje PCLXL para descripción de páginas se crea un objeto de la clase PCLXLSpool para el análisis del contenido PCLXL del fichero de spool.

Gráfico de llamadas para esta función:

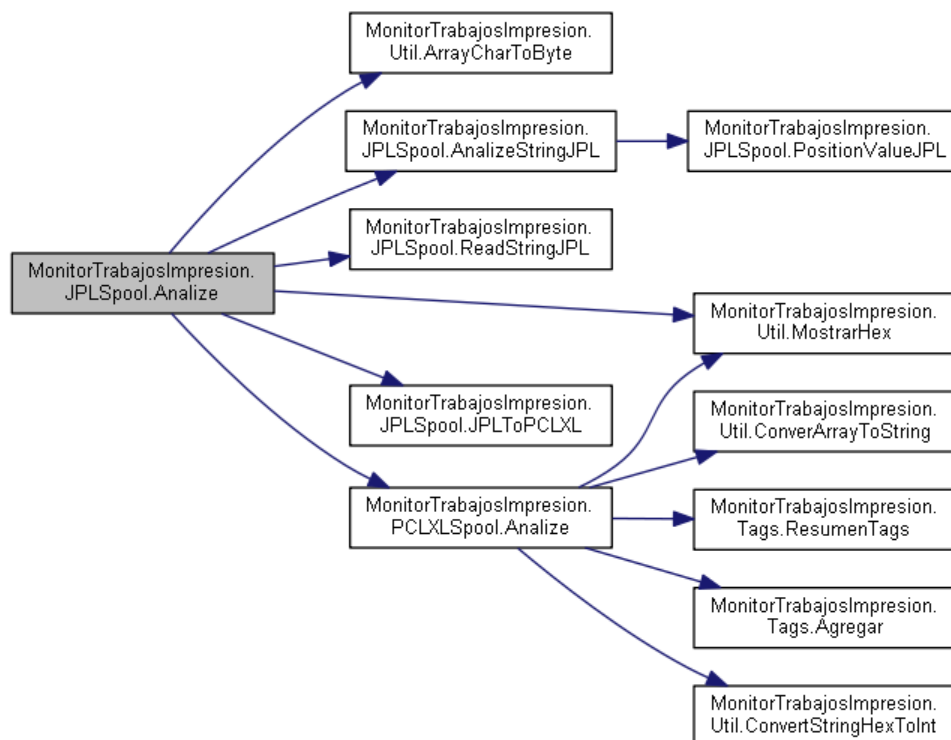


Ilustración 42. Gráfico de llamadas del método JPLSpool.Analyze

4.3.7.2.2 void AnalizeStringJPL (string *StringJPL*, ref STRUCT_PRINT_JOB *PrintJob*) [private]

Analiza la cadena JPL recibida como parámetro y si es necesario se actualizan las propiedades del Trabajo de Impresión.

Parámetros:

<i>StringJPL</i>	cadena JPL a analizar.
<i>PrintJob</i>	struct por referencia con las propiedades del trabajo de impresión

Seguimos la siguiente lógica:

- Comprobamos si la cadena JPL a analizar contiene alguna propiedad relevante del Trabajo de impresión. Si es así:
 - Obtenemos el valor de la propiedad del Trabajo de Impresión
 - Guardamos el valor de la propiedad del Trabajo de Impresión en la estructura pasada por referencia con las propiedades del trabajo de impresión.

4.3.7.2.3 bool JPLToPCLXL (string *StringJPL*) [private]

Analiza la cadena JPL que se pasa como parámetro para determinar si se cambia de JPL a lenguaje PCLXL.

Parámetros:

<i>StringJPL</i>	cadena JPL a analizar.
------------------	------------------------

Devuelve:

Verdadero si se cambia a lenguaje PCLXL

👁 *Si la cadena JPL de entrada, quitándole espacios en blanco y final de línea, coincide con esta cadena "@PJLENTERLANGUAGE=PCLXL", significa que se conmuta a lenguaje PCLXL*

4.3.7.2.4 `int PositionValueJPL (string StringJPL, int InitialPosition) [private]`

Obtiene la posición del valor de una instrucción JPL dentro de una cadena JPL (se ignoran los espacios en blanco e identificadores de comienzo de valor como ":" y "=").

Parámetros:

<i>StringJPL</i>	cadena JPL a analizar.
<i>InitialPosition</i>	Posición inicial en la que comenzar a buscar el valor del atributo dentro de la cadena JPL

Devuelve:

Posición real donde comienza el Valor del atributo dentro de la cadena JPL

4.3.7.2.5 `string ReadStringJPL (int PosicionJPL, ref BinaryReader JPLFile) [private]`

Devuelve la instrucción JPL localizada en la posición de fichero indicada en el parámetro entrada.

Parámetros:

<i>PosicionJPL</i>	Indica posición donde comienza el bloque JPL
<i>JPLFile</i>	con contenido JPL/PCLXL

4.3.7.3 Documentación de los datos miembro

Comando UEL para determinar comienzo y fin de fichero de Spool JPL y conmutación a otros lenguajes de impresión.

`string _COMANDO_UEL = (Convert.ToChar(Util.ESC) + "%-12345X") [static], [private]`

Comando JPL. Identifica el comienzo de una instrucción JPL.

`string _JPL_COMANDO = "@PJL" [const], [private]`

Comienzo de bloque PCLXL dentro de fichero de spool.

`int BeginBlockPCLXL [private]`

Fin bloque PCLXL dentro de fichero de spool.

`int EndBlockPCLXL [private]`

Logger para generación de logs: JPLSpool.

readonly ILog Log = LogManager.GetLogger("JPLSpool") [static], [private]

Para poder determinar si el análisis ha terminado.

bool .AnalisisPCLXLCompletado [get], [set]

Para poder determinar si el fichero de spool tiene contenido PCLXL.

bool JPLSpool.ContenidoPCLXL [get], [set]

4.3.8 Clase PCLXLSpool

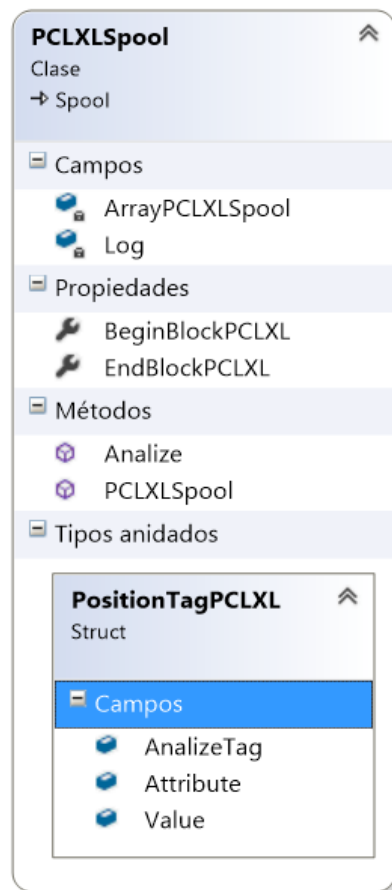


Ilustración 43. Clase PCLXLSpool.

Hereda de la clase Spool. Implementa el análisis de ficheros de **Spool** con contenido PCLXL

4.3.8.1 Documentación del constructor

PCLXLSpool (string PathFile, int IBeginBlockPCLXL, int IEndBlockPCLXL)

Inicializa atributos de clase y carga el fichero de spool con contenido PCLXL en un array.

Parámetros:

<i>PathFile</i>	Path del fichero de Spool
<i>IBeginBlockPCLXL</i>	Posición inicial del bloque PCLXL en el fichero de Spool
<i>IEndBlockPCLXL</i>	Posición final del bloque PCLXL en el fichero de Spool

Se sigue la siguiente lógica:

- Inicializa los atributos de clase con la posición inicial y final del bloque PCLXL del fichero de spool.
- Se abre el fichero de spool
- Se carga el fichero en un array
- Se cierra el fichero de spool.

4.3.8.2 Documentación de las funciones miembro

4.3.8.2.1 new void Analyze (ref STRUCT_PRINT_JOB *PrintJob*)

Análisis del contenido PCLXL del fichero de spool.

Parámetros:

<i>PrintJob</i>	struct con las propiedades del trabajo de impresión
-----------------	---

Se sigue la siguiente lógica:

Lectura de la cabecera del bloque PCLXL

Bucle para analizar los TAGS PCLXL del Spool. Se recorre todo el bloque PCLXL.

Leemos el siguiente TAG PCLXL a Analizar. Se pueden presentar los siguientes casos:

El Tag es un DataType (Tipo de Dato).

Los DataTypes pueden ser de dos grupos: Simple o Arrays. Para cada uno se implementa una lógica distinta para obtener (queda explicado en el título dedicado al estado del arte donde se analiza la estructura de los ficheros de spool con contenido PCLXL):

- El valor
- La posición del atributo al que está asociado.

Se obtiene el atributo al que corresponde el tipo de dato con su correspondiente valor.

Si el atributo está relacionado con el color de impresión se analiza su contenido y se guarda información, para la página actual, si es en color o b/n.

Si el atributo contiene información relevante del trabajo de impresión se actualizan las propiedades del trabajo de impresión:

DuplexPageMode: Impresión a una cara o a doble cara.

Orientation: Vertical u Horizontal.

PageCopies: Se incrementa con el valor del atributo el número de hojas.

MediaSize: Tamaño de papel.

ColorSpace: El documento es en b/n o puede tener contenido en color.

El Tag es un EmbedDataDefiner (Bloque de datos binarios).

Unicamente se obtiene el tamaño que ocupa el bloque binario para poder calcular la posición del siguiente Tag. No necesitamos analizar el contenido del bloque binario.

El TAG es un Operator

Si el operador indica una página nueva (BeginPage) para la nueva página se establece, en principio, que no tiene contenido en color.

Si el operador indica que se va a representar una imagen (BeginImage) se establece para la página actual que tiene contenido en color.

Se genera el log de información del operador y sus atributos.

Si no hay más Tag PCLXL se muestra un resumen con los Tags utilizados.

4.3.8.3 Documentación de los datos miembro

Posición inicial con contenido PCLXL en el fichero de **Spool**.

int BeginBlockPCLXL [get]

Posición final con contenido PCLXL en el fichero de **Spool**.

int EndBlockPCLXL [get]

Array en el que se guardan bloques de datos binarios del fichero de **Spool** para su análisis.

byte [] ArrayPCLXLSpool [private]

Se crea el Logger con nombre: PCLXLSpool.

readonly ILog Log = LogManager.GetLogger("PCLXLSpool") [static], [private]

4.3.9 Clase WatchIO

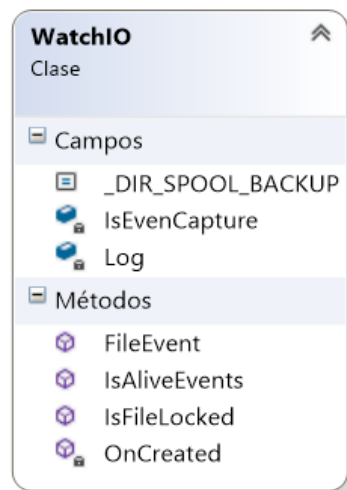


Ilustración 44. Clase WatchIO.

Realiza backup de los ficheros de spool generados por el Sistema de Impresión Windows.

4.3.9.1 Documentación del constructor

Sin constructor.

4.3.9.2 Documentación de las funciones miembro

4.3.9.2.1 static void MonitorTrabajosImpresion.WatchIO.FileEvent (string Path) [static]

Suscribe a los eventos del Sistema de archivos el método de clase OnCreated.

Al método se le aplica el siguiente atributo para asegurarnos que se tienen privilegios suficientes para realizar las operaciones requeridas en el Sistema de Archivos.

```
// Atributo para asignar obtener privilegios necesarios en el sistema de archivos  
[PermissionSet(SecurityAction.Demand, Name = "FullTrust")]  
public static void FileEvent(string Path)
```

Detalle 40. Clase WatchIO. declaración método FileEvent

Parámetros:

Path	Directorio de Spool de la cola de Impresión.
------	--

Se sigue la siguiente lógica:

- Comprobamos si existe el directorio de backup para Spool de impresión. Si no existe se crea.
- Creación del evento del sistema de archivos que supervisa el directorio de Spool de impresión
- Supervisamos cambios, en tiempos de último acceso y última escritura, y renombrado de ficheros y directorios
- Sólo se supervisan en el directorio los ficheros de Spool.
- Suscribimos el método delegado, OnCreated, al evento de creación de fichero.
- Se actualiza el atributo de clase IsEvenCapture a verdadero (acabamos de suscribirnos para capturar eventos de creación de archivos)

- Actualizamos la variable de control de suscripción para chequear periódicamente si estamos suscritos a los eventos del Sistema de Archivos.
- Activamos la suscripción al evento.

4.3.9.2.2 `static void IsAliveEvents (object sender, EventArgs Args) [static]`

Testea si seguimos suscritos a los eventos del Sistema de Archivos. Esta función se lanza periódicamente por que está suscrita a un objeto Timer.

Parámetros:

No se comentan por que son parámetros por defecto para suscribir un método estático a los eventos de un objeto de la clase Timer y no se utilizan.

Se sigue la siguiente lógica:

- Se actualiza el atributo de clase `IsEvenCapture` como falso.
- Se crea el fichero de test: TEST.SPL, sin contenido. Después de una pausa se borra el fichero de test. (si se está suscrito debe haberse producido el evento de creación de fichero y ejecutarse el método suscrito `OnCreated` por lo que el atributo de clase debe devolver verdadero)
- Se chequea el atributo de clase `IsEvenCapture` y si devuelve falso volvemos a suscribirnos lanzando el método `FileEvent` y actualizar el atributo de clase a verdadero.

4.3.9.2.3 `static bool IsFileLocked (FileInfo file) [static]`

Con este método comprobar si un fichero está bloqueado.

Parámetros:

<i>FileInfo</i>	Contiene las propiedades de un fichero.
-----------------	---

Devuelve:

Valor booleano indicando si el fichero está bloqueado o no.

Se sigue la siguiente lógica:

- Se intenta abrir el fichero que se pasa como parámetro en modo exclusivo.
- Si da una excepción de E/S de tipo `IOException` consideramos que el fichero permanece bloqueado y devuelve falso.

4.3.9.2.4 `static void OnCreated (object source, FileSystemEventArgs edata) [static], [private]`

Método delegado suscrito al evento de creación de archivo con el que se realiza una copia del fichero de Spool anexando la fecha y hora de creación a la copia del fichero.

Parámetros:

<i>source</i>	instancia de objeto que lanza el evento
---------------	---

<i>edata</i>	datos para los eventos de directorio
--------------	--------------------------------------

Se sigue la siguiente lógica:

- El nombre del fichero de backup se obtiene poniéndole como sufijo, al nombre del fichero original, el nombre, fecha y hora.
- Si se ejecuta este método es por que está suscrito a los eventos del Servidor de Impresión por lo que establecemos con valor verdadero el atributo de clase `IsEvenCapture`.
- Se chequea que el fichero de spool no esté bloqueado.
- Se hace una copia del fichero de Spool

4.3.9.3 Documentación de los datos miembro

Subdirectorio para backup de los ficheros de spool relativo al directorio de **Spool**.

```
const string _DIR_SPOOL_BACKUP = "MonitorImpresion"
```

Variable para control de subscripción Eventos.

```
bool IsEvenCapture = false [static], [private]
```

Se crea el Logger con nombre: WatchIO.

```
readonly ILog Log = LogManager.GetLogger("WatchIO") [static], [private]
```

4.3.10 Clase Programa

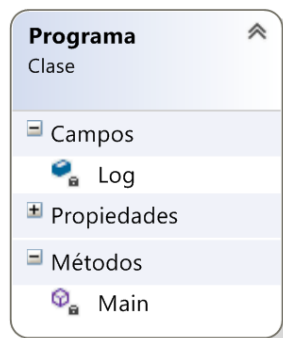


Ilustración 45. Clase Programa.

Contiene el método `Main()` con el que se inicia el proceso para monitorizar trabajos de impresión.

4.3.10.1 Documentación del constructor

Sin constructor.

4.3.10.2 Documentación de las funciones miembro

4.3.10.2.1 static void Main () [static], [private]

Punto de entrada al Servicio Monitor de Impresión.

Se sigue la siguiente lógica:

- Leemos la configuración en el fichero de configuración de la aplicación para log4net.
- Inicializamos log4net para que recargue la configuración en tiempo de ejecución.
- Se configuran las impresoras locales para que conserven los trabajos después de su impresión.
- Se crea una instancia de la clase ServicePrintMonitor para que se ejecute como Servicio de Windows.

4.3.10.3 Documentación de los datos miembro

Se crea el Logger con nombre: Programa.

readonly ILog Log = LogManager.GetLogger("Programa") [static], [private]

4.3.11 Clase ServicePrintMonitor

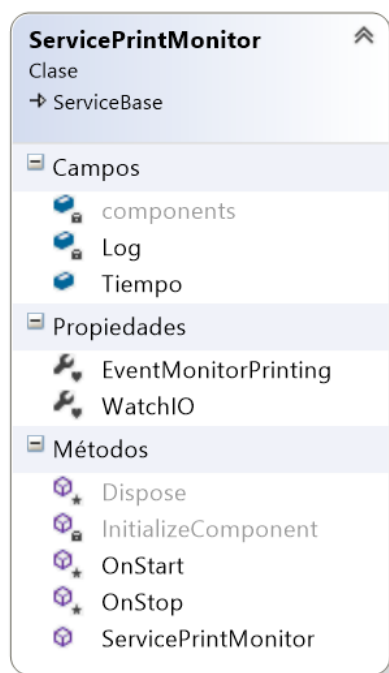


Ilustración 46. Clase ServicePrintMonitor.

Servicio para activar el monitor de impresión en segundo plano y mantenerlo en ejecución hasta que no se detenga el servicio. Hereda de la clase ServiceBase.

4.3.11.1 Documentación del constructor

MonitorTrabajosImpresion.ServicePrintMonitor.ServicePrintMonitor ()

Gráfico de llamadas a esta función:

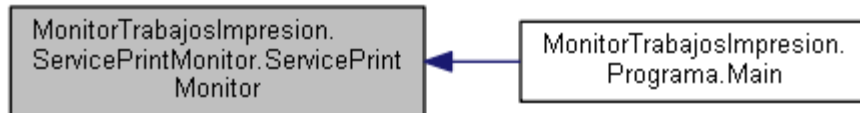


Ilustración 47. Gráfico de llamadas al constructor de la clase ServicePrintMonitor.

Se sigue la siguiente lógica:

- Se inicializa el servicio.
- Nos suscribimos a los eventos de impresión del registro de Eventos de Windows.
- Nos suscribimos a los eventos del Sistema de archivos asociados al directorio de spool de la cola de impresión.
- Se define un Timer para comprobar periódicamente que seguimos suscritos a los eventos de impresión y de creación de ficheros de spool.

4.3.11.2 Documentación de las funciones miembro

4.3.11.2.1 override void Dispose (bool disposing) [protected]

Limpiar los recursos que se estén utilizando.

Método necesario para admitir el Diseñador. No se puede modificar el contenido del método con el editor de código.

Parámetros:

<i>disposing</i>	true si los recursos administrados se deben eliminar; false en caso contrario, false.
------------------	---

4.3.11.2.2 void InitializeComponent () [private]

Método necesario para admitir el Diseñador. No se puede modificar el contenido del método con el editor de código.

4.3.11.2.3 override void OnStart (string[] args) [protected]

Se habilita el atributo de clase Tiempo (clase Timer).

4.3.11.2.4 override void OnStop () [protected]

Sin contenido.

4.3.11.3 Documentación de los datos miembro

Variable del diseñador requerida.

System.ComponentModel.IContainer components = null [private]

Se crea el Logger con nombre: ServicePrintMonitor.

```
readonly ILog Log = LogManager.GetLogger("ServicePrintMonitor") [static],  
[private]
```

Timer utilizado para revisar que seguimos suscritos a los eventos de impresión y de creación de ficheros en el directorio de spool de la cola de impresión.

Timer Tiempo

4.3.12 Interface IDB



Con WCF vamos a crear un servicio, alojado en un servidor IIS, con el que poder interactuar con nuestra BD. Este servicio va a implementar la capa CAD.

Para ello se crea la interfaz IDB con la que voy a exponer el servicio (método EjecutaSQL). La interfaz IDB se implementará en la clase DB.

Definimos el contrato de servicio aplicando el atributo [ServiceModelContract] a la interfaz IDB. Y para exponer los métodos como servicios les tenemos que aplicar el atributo [OperationContract].

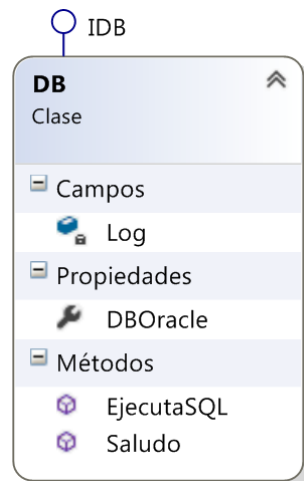
Para crear servicios con WCF es necesario utilizar el namespace System.ServiceModel.

El siguiente código muestra la implementación de nuestra interface.

```
using System.ServiceModel;  
namespace WCFDB  
{  
    [ServiceContract]  
    public interface IDB  
    {  
        [OperationContract]  
        string Saludo(string value);  
  
        [OperationContract]  
        bool EjecutaSQL(string SQL);  
    }  
}
```

Detalle 41. Interfaz de servicio WCF

4.3.13 Clase DB



Implementa la interfaz IDB.

4.3.13.1 Documentación del constructor

Sin constructor.

4.3.13.2 Documentación de las funciones miembro

4.3.13.2.1 bool WCFDB.DB.EjecutaSQL (string SQL)

Lanza una instrucción SQL a la BD.

Parámetros:

SQL	string con la sentencia SQL
-----	-----------------------------

Devuelve:

Un bool para indicar si ha sido correcta (true) o errónea la ejecución de la instrucción SQL recibida como parámetro.

Se sigue la siguiente lógica:

- Filtramos sentencias SQL no permitidas. Sólo inserciones en la tabla SPOOL_PRINTJOBS.
- Se crea un objeto de la clase DBOracle y se invoca el método para ejecutar sentencias SQL.

4.3.13.2.2 string WCFDB.DB.Saludo (string value)

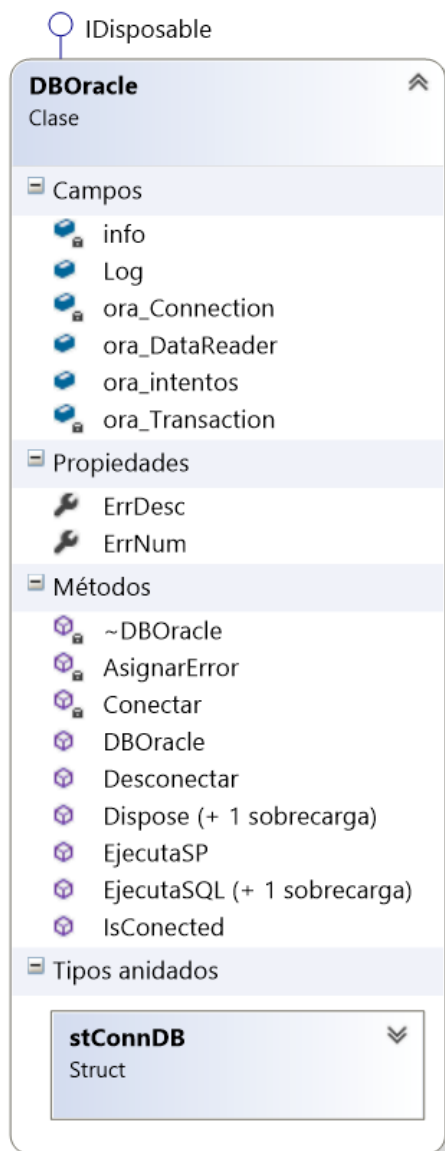
Método para testar que el Servicio Web está operativo.

4.3.13.3 Documentación de los datos miembro

Se crea el Logger con nombre: [Class: DB].

readonly ILog Log = LogManager.GetLogger("[Class: DB]")[static], [private]

4.3.14 Clase DBOracle



Esta clase se puede copiar y utilizar según (MuchoCodigo, n.d.). La he incluido en el namespace WCFDB tal cual está publicada salvo alguna ligera modificación para definir la cadena de conexión en BD y generación de logs.

Diagrama de colaboración para WCFDB.DBOracle:

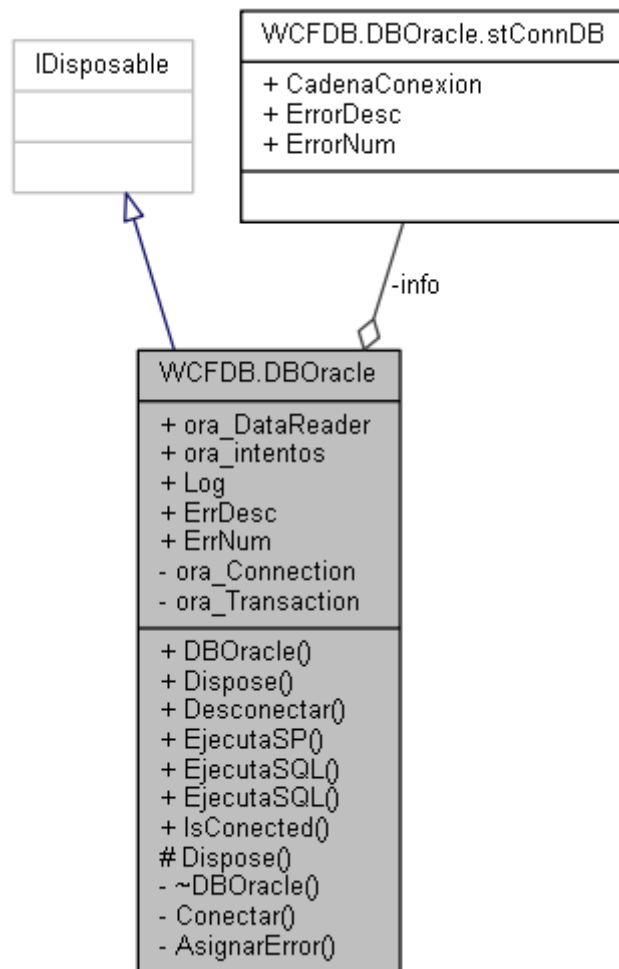


Ilustración 48. Diagrama de colaboración de la clase DBOracle.

A continuación una breve descripción.

4.3.14.1 Documentación del constructor y destructor

- `DBOracle ()`

Constructor. En la clase original tal como la compartía el autor se define la cadena de conexión a BD. Sin embargo, aunque no se ha modificado, ahora defino la cadena de conexión a BD en el archivo de configuración `App.Config`.

- `~DBOracle ()`

Destructor. Libera recursos administrados.

4.3.14.2 Documentación de las funciones miembro

- `bool Desconectar ()`

Cierra la conexión de BD.

- `bool EjecutaSP (ref OracleCommand OraCommand, string SpName)`

Ejecuta un procedimiento almacenado de Oracle.

- bool **EjecutaSQL** (string SqlQuery)

Ejecuta una sql para rellenar un DataReader (sentencia select).

- bool **EjecutaSQL** (string SqlQuery, ref int FilasAfectadas)

Ejecuta una sql que no devuelve datos (update, delete, insert).

- bool **IsConected** ()

Devuelve el estado de la base de datos

- void **Dispose** ()

Implementa IDisposable.

- virtual void **Dispose** (bool disposing)

Dispose de la clase.

- bool **Conectar** ()

Se conecta a una base de datos de Oracle.

- void **AsignarError** (ref Exception ex)

Captura Excepciones

4.3.14.3 Documentación de los datos miembro

- static readonly ILog **Log** = LogManager.GetLogger("DBOracle")

Se crea el Logger con nombre: DBOracle.

- string **ErrDesc** [get]

Devuelve la descripción de error de la clase.

- string **ErrNum** [get]

Devuelve el número de error de la clase.

- OracleConnection **ora_Connection**

Conexión a BD.

- OracleTransaction **ora_Transaction**

Transacción Oracle.

- string **stConnDB info**

Cadena de conexión a BD.

4.4 Implantación con SCRUM

Me encuentro a principios de abril (2016). Ya tengo implementado el Servicio Monitor de Impresión y la Capa CAD, es el momento de probar su funcionamiento en un entorno real para ver si cumple con los requisitos, detectar nuevos requisitos, detectar y corregir errores.

Para hacer esta planificación he utilizado SCRUM por los siguientes motivos:

- Conseguir mayor implicación de los usuarios finales al que se destina este trabajo haciéndolos partícipes también.
- las pruebas realizadas no son lo suficientemente fiables y en los trabajos capturados hay información que habría que contrastar en un entorno real.
- Poder planificar en el tiempo la fase final del TFG y decidir que producto o partes del producto están finalizadas.

4.4.1 Reunión de planificación del proyecto.

La reunión tiene lugar el día 5/4/2016.

En ella intervienen:

1. Como Scrum Owner: Encarnación García, Jesús Rubén García y Ana López. Por su implicación directa con el uso de la aplicación a desarrollar.
2. Como Scrum Master: José Aguilera, Julio López y Andrés Martínez.
3. Como Scrum Team el que suscribe este proyecto.

La composición de cada uno de los roles, en cuanto al número de miembros, no está dentro de las recomendaciones de SCRUM, pero como estas no son imperativas, he creído muy conveniente y enriquecedor no tenerlas en cuenta y contar con todos ellos por su relación con este trabajo, sus conocimientos y su valiosa experiencia.

En la reunión se tratan los siguientes puntos:

4.4.1.1 Duración de las iteraciones

Se establece en 2 semanas (10 días hábiles, se excluyen sábados y domingos) para la duración de las iteraciones, de este modo se pueden hacer correcciones más rápidamente en cuanto a estimaciones de tiempo, modificar funcionalidades, revisar prioridades, corregir la VT o en caso necesario desechar parte del trabajo realizado durante la iteración. Así podemos ajustar mejor la planificación.

4.4.1.2 Velocidad de trabajo (VT)

La VT por iteración se establece de la siguiente manera utilizando un factor de corrección de 0,6 y suponiendo 8 horas de trabajo al día:

$VT = 10 \text{ días} * 8 \text{ horas/día} * 0,6 = 48 \text{ horas por iteración.}$

Redondeando podemos asumir 50 horas por iteración.

4.4.1.3 Delimitar el ámbito de la implantación

Se acuerda centrarnos en implantar el Servicio Monitor de Impresión para los dispositivos de impresión para los que la aplicación Megatrack no puede capturar trabajos de impresión, es decir, para los plotters. De esta manera no se interfiere con el trabajo que ya se está realizando.

Posterior a este TFG se realizará la aplicación para explotar la información y poder calcular los costes de impresión, así como el resto de funcionalidades.

Llegados a este punto se estará en disposición de reemplazar el actual sistema Megatrack con el menor impacto posible para el resto de dispositivos de impresión.

La siguiente ilustración muestra el diagrama de componentes a implantar como objetivo de este TFG:

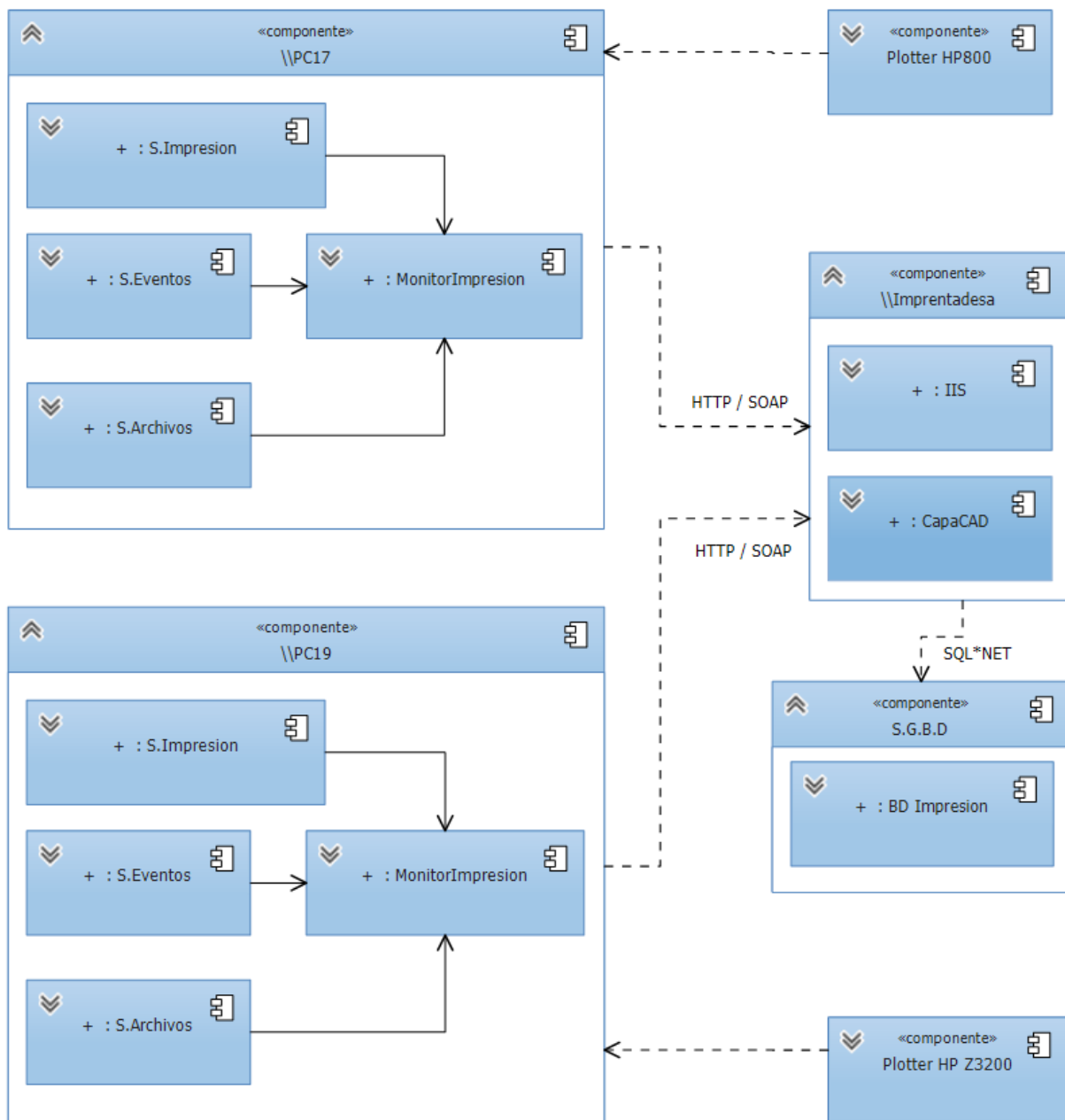


Ilustración 49. Diagrama de componentes del sistema a implantar.

4.4.1.4 Funcionalidades (Product Backlog).

Funcionalidades solicitadas por los Scrum Owner.

ID	Funcionalidad	Prior	Horas	Como probarlo	Notas
F.1	Controlar impresión por usuario.	500	40	Revisar a través de los logs que los trabajos de impresión se han guardado en B.D.	Recuperar para cada trabajo de impresión los siguientes detalles: ✓ Usuario: Se obtiene el usuario con que inicia sesión. ✓ Número hojas totales

				Revisar que los trabajos de impresión apuntados por los usuarios se han registrado en BD.	<ul style="list-style-type: none"> ✓ Número hojas en color. ✓ Número de copias. ✓ Tamaño papel: Parametrizado (P.e: A3, A0, etc) o personalizado (P.e: 1200 x 914 mm). ✓ Impresión en color o b/n ✓ Fecha impresión. ✓ Nombre del dispositivo de impresión. ✓ Tipo dispositivo de impresión: Plotter o impresora. ✓ Servidor impresión. ✓ Nombre de documento ✓ Impresión simple o a doble cara. ✓ Orientación del papel (vertical, horizontal) ✓ Tipo de papel (para plotter).
F.2	Revisar y validar la información obtenida de los trabajos de impresión.	500	10	Habilitando una pantalla a los operadores donde se muestran los trabajos de impresión y pueden hacer anotaciones y validar su corrección.	Se debe garantizar que la información obtenida de los trabajos de impresión es correcta.
F.3	Facturar lo impreso por usuario (Costes).	400			<p>Establecer costes de impresión:</p> <p>- Plotter:</p> <ul style="list-style-type: none"> ✓ En función del tamaño de impresión se toma como referencia el tamaño A0. ✓ Tipo de papel (el operador configura el tipo de papel antes de imprimir) <p>- Impresoras:</p> <ul style="list-style-type: none"> ✓ En función del tamaño de papel. A3 y A4. Todo lo que no sea A3 se considera A4.
F.4	Generar listados con gastos de impresión.	300			<p>4. Listado por registro electrónico AAEE</p> <p>5. Listado por email al</p>

					responsable unidad de gasto
F.5	Tratamiento incidencias.	200			Impresiones erróneas o sin finalizar debido a: ✓ Impresora sin papel. ✓ Plotter sin suficiente papel ✓ Se acaba la tinta o tóner antes de acabar la impresión. ✓ Etc.
F.6	Identificar cantidad de tinta utilizada en un trabajo de impresión	100			
F.7	Contabilizar gastos en consumibles	100			<ul style="list-style-type: none"> - Dispositivos impresión - Tinta - papel
F.8	Control de centro de gasto	100			El usuario tiene que indicar el centro de gasto antes de imprimir.

Tabla 35. Scrum. Product Backlog

4.4.2 Primera iteración

4.4.2.1 Sprint planning Meeting (Reunión de la iteración): Objetivos

El objetivo de la primera iteración es completar las dos primeras funcionalidades: F.1 y F.2, es decir, asegurar que se capturan los trabajos de impresión y garantizar que los datos obtenidos son correctos.

Cumpliendo los objetivos de la primera iteración estaría cubierto el trabajo que queda pendiente de este TFG: implantación y pruebas.

4.4.2.1.1 Elementos de trabajos para cada Funcionalidad

4.4.2.1.1.1 F.1 Controlar impresión por usuario.

Product Backlog ID: F.1	Prioridad	Estimación
Proyecto: Monitor Impresión	500	35
Notas		
<p>Recuperar para cada trabajo de impresión los siguientes detalles:</p> <ul style="list-style-type: none">✓ Usuario: Se obtiene el usuario con que inicia sesión.✓ Número hojas totales✓ Número hojas en color.✓ Número de copias.✓ Tamaño papel: Parametrizado (P.e: A3, A0, etc) o personalizado (P.e: 1200 x 914 mm).✓ Impresión en color o b/n✓ Fecha impresión.✓ Nombre del dispositivo de impresión.✓ Tipo dispositivo de impresión: Plotter o impresora.✓ Servidor impresión.✓ Nombre de documento✓ Impresión simple o a doble cara.✓ Orientación del papel (vertical, horizontal)✓ Tipo de papel (para plotter).		
Como probarlo		
<p>Revisando las propiedades de los trabajos de impresión capturadas en:</p> <ul style="list-style-type: none">- Log del servicio Monitor de Impresión- Log del <u>WebService</u> de la capa CAD.- SQL consultando la BD donde se registran los trabajos de impresión.- Hojas donde los usuarios apuntan los trabajos de impresión que realizan		

Elementos de Trabajo asociados a la funcionalidad F.1

ID	Tarea	Estimación (Horas)
E.T 1.1	Automatizar pruebas. Pruebas de sistema.	15

E.T 1.2	Refactorizar	15
E.T 1.3	Añadir propiedades: Es plotter y tipo de papel.	5

Tabla 36. E.T asociados a funcionalidad F.1

4.4.2.1.1.2 F.2 Revisar y validar la información obtenido de los trabajos de impresión.

Product Backlog ID: F.2
Proyecto: Monitor Impresión

Prioridad

500

Estimación

15

Notas

Se debe garantizar que la información obtenida de los trabajos de impresión es correcta.

Como probarlo

Habilitando una pantalla a los operadores donde se muestran los trabajos de impresión y puedan hacer anotaciones y validar su corrección.

Elementos de trabajo de la funcionalidad F.2

ID	Tarea	Estimación (Horas)
E.T 2.1	Crear Formulario Access para revisar trabajos de impresión.	3
E.T 2.2	Webform para que los operadores revisen los trabajos de impresión.	12
E.T 2.3	Revisión resultados obtenidos.	10
E.T 2.4	Instalación Servicio Monitor Impresión en equipos con plotter.	2

Tabla 37. E.T asociados a funcionalidad F.2

4.4.2.2 Desarrollo iteración

La siguiente ilustración muestra la evolución de la iteración a través de su gráfica burndown.

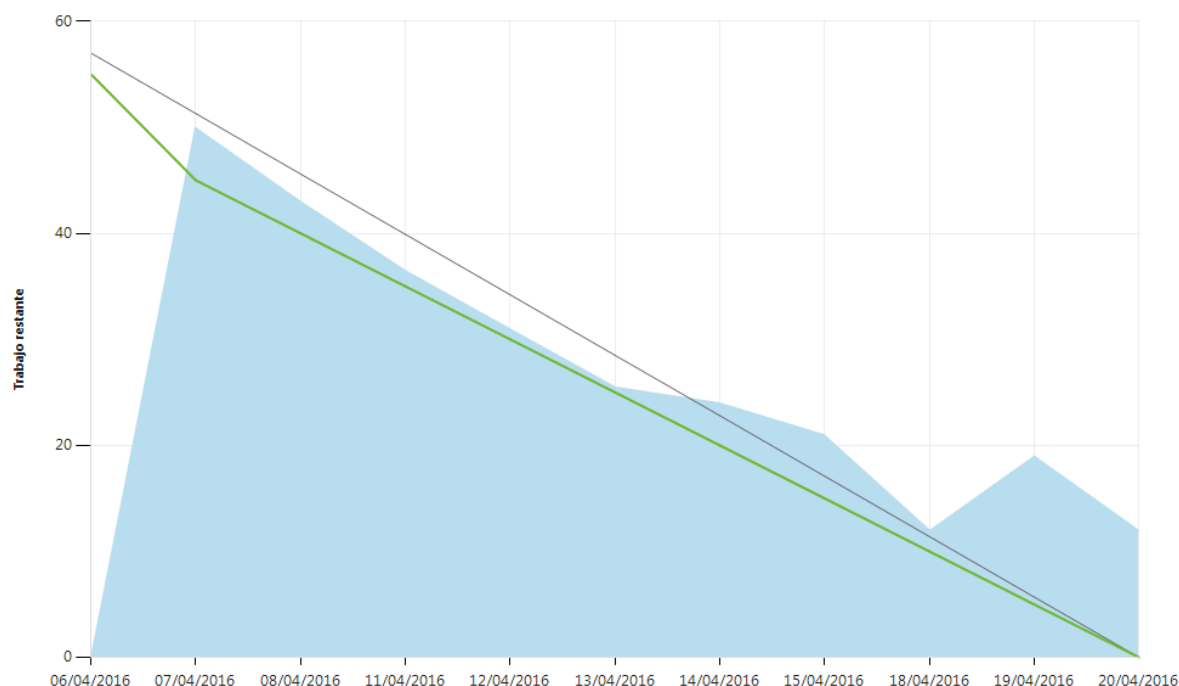


Ilustración 50. Gráfica burndown de la 1ª iteración.

Los desajustes que se observan en la fase inicial de la iteración es debido a la falta de manejo de la herramienta Visual Studio Team Foundation ya que son los primeros contactos con la herramienta.

El desajuste final es debido a que se decide descartar la tarea E.T 2.2 pues la tarea E.T 2.1 ya cubre ese mismo trabajo y se crea la nueva tarea E.T 2.3 para revisar los resultados obtenidos.

4.4.2.2.1 E.T 2.4 Instalación Servicio Monitor Impresión en equipos con plotter.

E.T 2.4 Instalación Servicio Monitor Impresión en equipos con plotter

2

T 2.4.1 Instalar
Servicio Monitor de
Impresión

1

T 2.4.2
Configuración
Servicio Monitor
Impresión

1

Ilustración 51. Tareas de E.T 2.4

👁 En los elementos de trabajo y las tareas se indica las horas de trabajo estimadas

Se instala el Servicio Monitor de impresión en los dos equipos que tienen plotter conectado.

Se requieren las dos siguientes tareas.

T 2.4.1: La instalación no tiene mayor complicación utilizando el instalador que se ha creado en el proyecto con Visual Studio. En la siguiente ilustración se muestran una serie de capturas con la instalación:

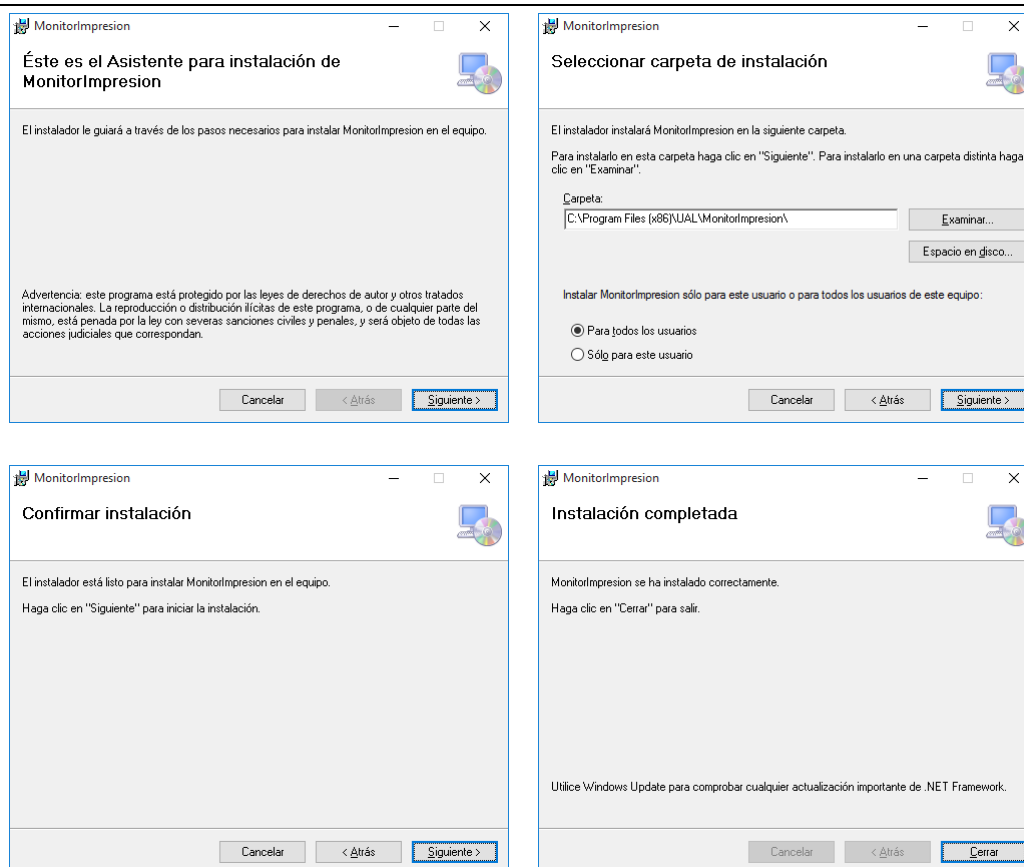


Ilustración 52. Instalación del Servicio Monitor de Impresión.

T 2.4.2: Tras la instalación podemos acceder a los servicios de Sistema para comprobar que el Servicio se ha instalado correctamente y modificar su configuración para que se inicie automáticamente como se muestra en la siguiente ilustración:

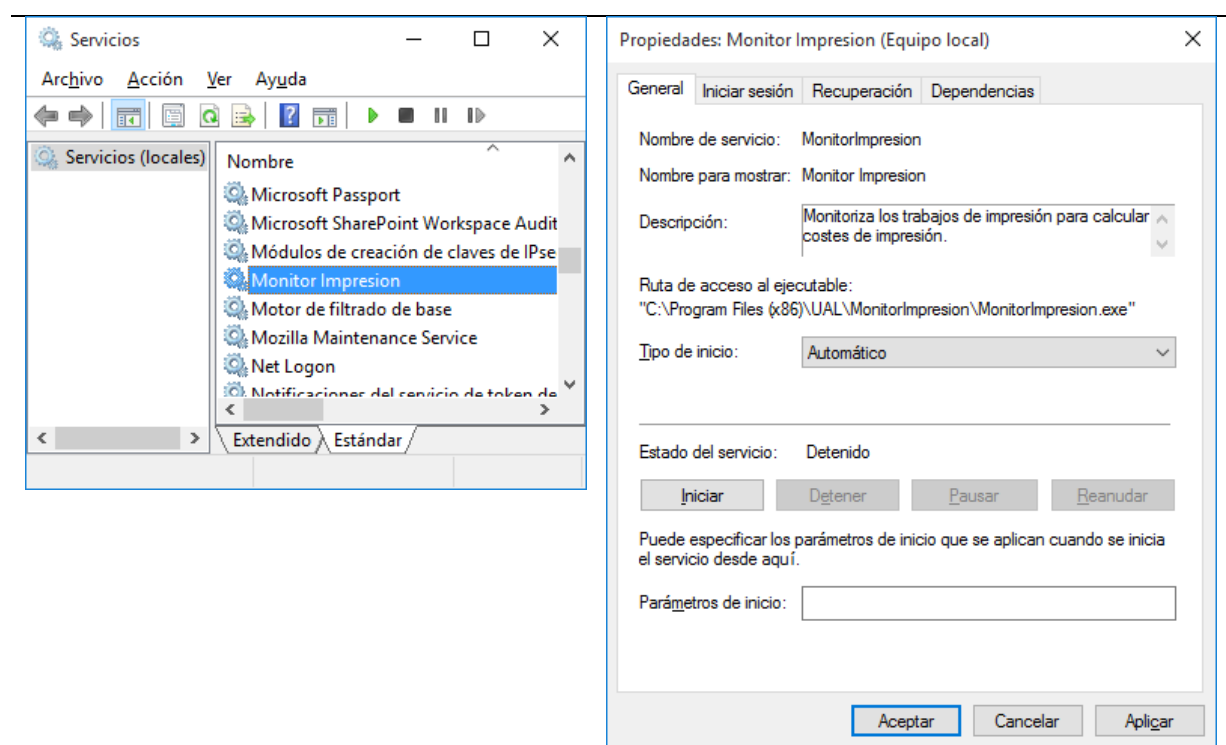


Ilustración 53. Servicio Monitor de Impresión.

4.4.2.2.2 E.T 2.1 Crear Formulario Access para revisar trabajos de impresión.

E.T 2.1 Crear Formulario Access para revisar trabajos de impresión

4

T2.1.1 Instalar cliente Oracle y configurar ODBC

1

T2.1.2 Dar acceso a IP operadores a BD Oracle

1

T2.1.3 Crear BD Access con Formulario para consultar trabajos de impresión.

1

T2.1.4 Modificar BD, estructura datos y código para nuevos campos.

1

Ilustración 54. Tareas de E.T 2.1

El objetivo a llevar a cabo con el elemento de trabajo E.T 2.1 es poner a disposición de los operadores un formulario Access donde poder consultar los trabajos de impresión almacenados en BD y poder hacer anotaciones.

Se ha requerido de tres tareas:

T 2.1.1: Para poder acceder a la BD Oracle desde Access es necesario tener instalado un cliente Oracle que incluya el ODBC para Oracle. Para ello se ha utilizado un cliente Oracle gratuito que incluye ODP.NET y el ODBC y se ha creado un ODBC de sistema para poder conectar con la BD Oracle. En el anexo se dan más detalles sobre el cliente Oracle.

T 2.1.2: También se ha tenido que habilitar la IP de los PCs de los operadores para que puedan acceder al SGBD donde se encuentra la BD del Monitor de Impresión.

T 2.1.4: Se modifica la tabla SPOOL_PRINTJOBS para añadir los campos ID_REVISADO y ID_NOTAS, además se han tenido que revisar las estructuras de datos y codificación del Monitor de impresión. Estos nuevos campos se utilizan para que los operadores puedan hacer anotaciones sobre los trabajos de impresión.

T 2.1.3: Por último se ha creado una BD Access donde se ha vinculado la tabla Oracle SPOOL_PRINTJOBS y se ha creado un formulario para que los operadores puedan revisar los trabajos de impresión y hacer anotaciones. En la siguiente ilustración se muestra el formulario

PR_LOGIN_SPOOL_PRINTJOBS			
ID_JOBNAME	00012_07/04/2016_01:58:21	N_PAGES_PRINTED	1
F_PRINTJOB	07/04/2016 1:58:21	N_COLORPAGES	1
ID_FUENTE	API	N_LENGTH	2970
ID_LOGIN	Miguel	N_WIDTH	2100
N_JOB	12	ID_MEDIASIZE	DMPAPER_A4
ID_PRINTSERVER	\\TAMARIX	N_MEDIASIZE	9
ID_PRINTER	PDFCreator	ID_MEDIATYPE	
ID_ISPLOTTER		N_MEDIATYPE	1
ID_DOCUMENT	C:\Program Files (x86)\Tomcat 6.0\webapps\xplanner-plus\WEB-INF\classes\xplanner-custom.properties	ID_COLOR	Color
N_PAGES	1	N_COPIES	1
		N_JOBSIZE	0
		ID_STATUS	0
		ID_REVISADO	
		ID_NOTAS	

Ilustración 55. Formulario Access para consultar y hacer anotaciones en trabajos de impresión.

4.4.2.2.3 E.T 1.1 Automatizar pruebas. Pruebas de sistema.

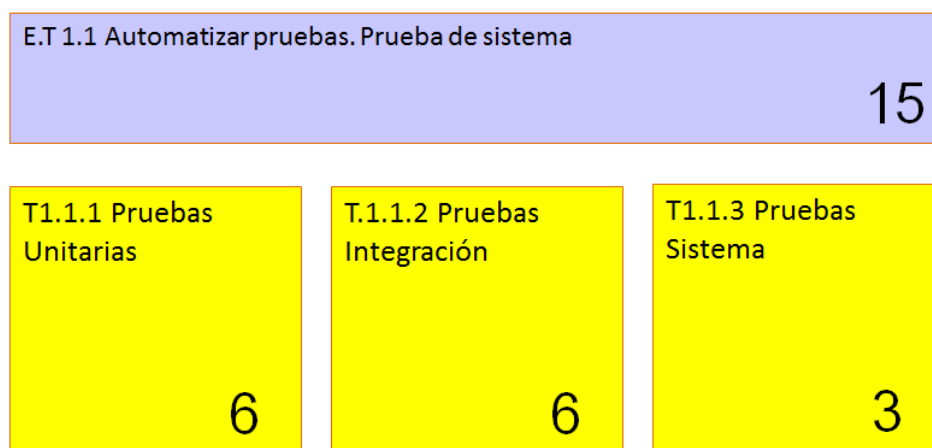


Ilustración 56. Tareas de E.T 1.1

Aunque durante las 2 primeras fases de desarrollo del Monitor de Impresión se han ido probando todos los componentes a medida que se iban desarrollando se ha visto conveniente aprovechar la facilidad de crear pruebas automatizadas que ayuden a detectar errores sobre todo para las sucesivas modificaciones.

Para ellos se han creado 3 tareas:

T 1.1.1: Se crean pruebas unitarias. Para probar la clase Util que contiene varios métodos utilizados para el análisis de ficheros de spool

T 1.1.2: Se crean pruebas de integración. Se prueba la lógica de las funcionalidades más importantes donde la prueba de un método lleva implícito el funcionamiento correcto de otros métodos y clases. Por ejemplo probar el análisis de lenguaje EMF, JPL y PCLXL en ficheros de Spool.

Para construir los test en las T 1.1.1 y T 1.1.2 se sigue el siguiente modelo con la herramienta de testeo en Visual Studio:

1. Inicialización (Arrange).
2. Ejecución (Act).
3. Comprobación (Assert).

Vamos a ver un test simple de los que he implementado; comentado para su comprensión:

```
/** \brief Test sobre la clase Util
 */
[TestClass] // Atributo para indicar que es una clase para test
public class UtilTest : Util
{
    /** \brief Prueba el método "ArrayCharToByte()".
     * \details Este método convierte un array de char (char[]) en un array de bytes (byte[]).
     * - Para probarlo definimos un string con un valor determinado y guardamos su representación en un array de char.
     * - Al array de char le aplicamos el método que estamos probando, ArrayCharToByte(), obteniendo su representación en un array de byte.
     * - Al array de byte le aplicamos el método ConverArrayToString(). El resultado
```

```

debe ser el string original
* /remarks Con esta prueba además del método "ArrayCharToByte()" probamos el
método "ConverArrayToString()"
*/
[TestMethod] // Atributo para indicar que es un método tipo test
// Los métodos no devuelven ningún valor (void)
public void Util_ArrayCharToByte_ConvierteArraDeCharEnArrayDeByte()
{
    /// Inicialización (Arrange)
    /// - Inicializamos un string
    /// - Inicializamos un char[] con la representación del string anterior en un
array de char
    string cadena = "Hola caracola";
    char[] arrayChar = cadena.ToCharArray();

    /// Ejecución (Act)
    /// - Ejecutamos el método a probar, "ArrayCharToByte()", con el char[] y
obtenemos un byte[]
    /// - Convertimos el byte[] en un string
    byte[] arrayByte = ArrayCharToByte(arrayChar);
    string cadenaDesdeArrayByte = ConverArrayToString(arrayByte);

    /// Comprobación (Assert)
    /// Comprobamos que el contenido del string original y el obtenido tras la
ejecución es la mismo
    Assert.IsTrue(cadena.Equals(cadenaDesdeArrayByte));
}
}

```

Detalle 42. Ejemplo de codificación de prueba automatizada.

En la siguiente captura se muestra el resultado de las pruebas implementadas:

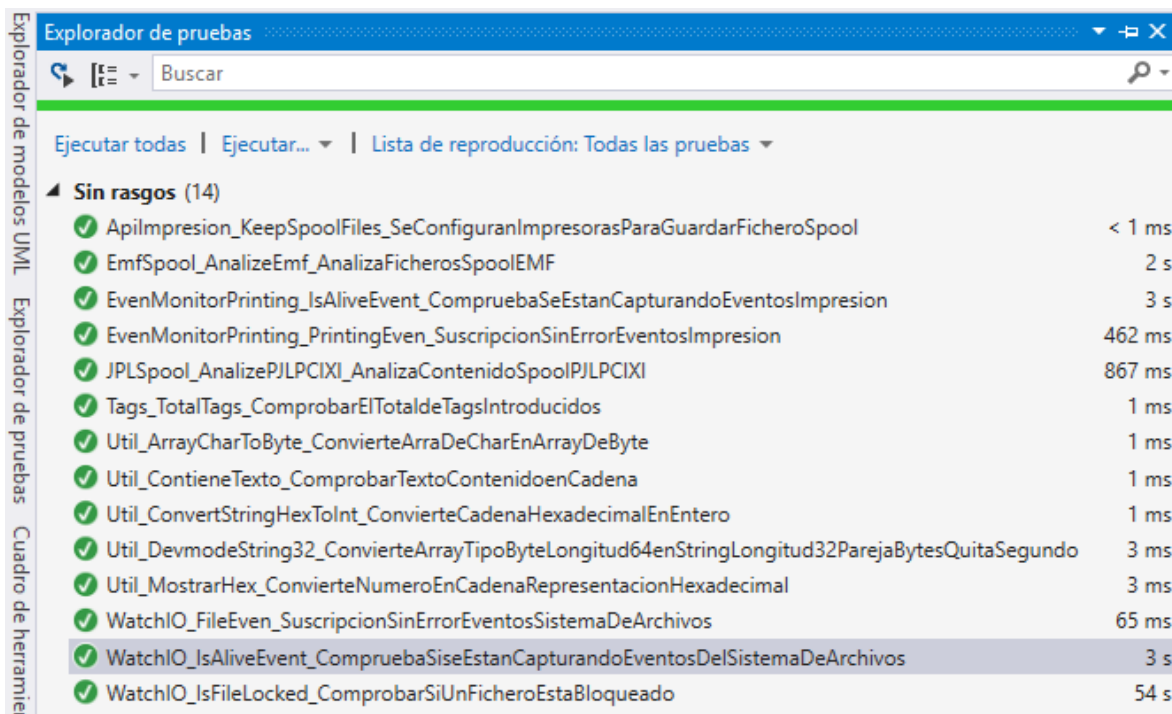


Ilustración 57. Captura de resultados de las pruebas automatizadas.

T 1.1.3: Pruebas de sistema. Se trata de probar que el conjunto de la aplicación funciona correctamente. Estas pruebas no están automatizadas y se tiene que comprobar que se obtienen

los resultados esperados a través de los logs del Servicio Monitor de Impresión, los logs del Webservice de la capa CAD y de las propiedades de los trabajos de impresión en BD.

4.4.2.2.4 E.T 1.2 Refactorizar

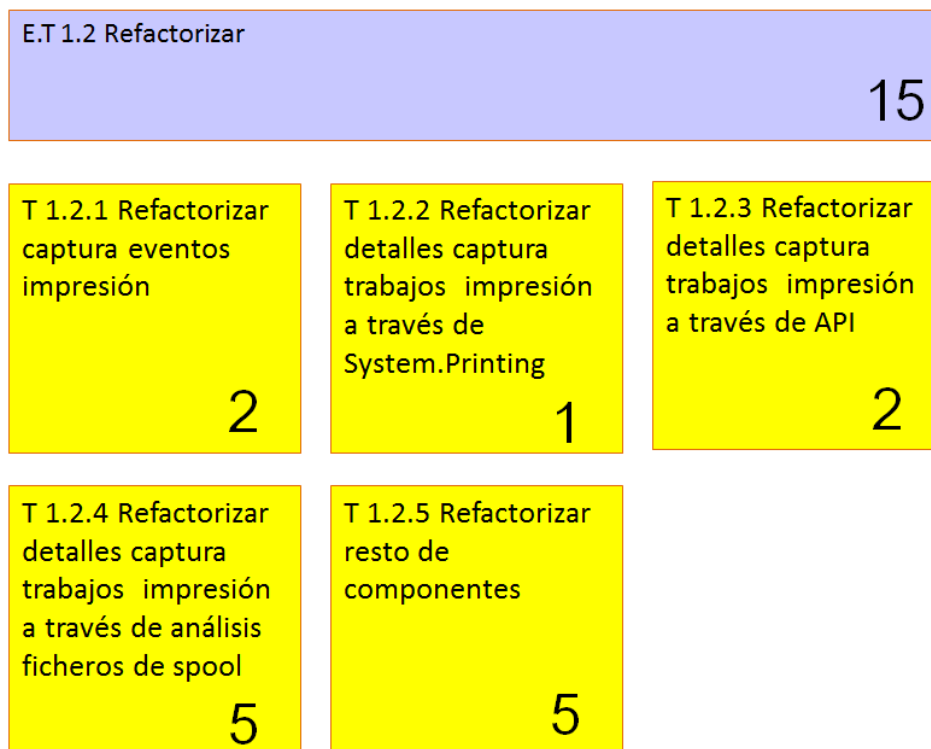


Ilustración 58. Tareas de E.T 1.2

Además de crear pruebas automatizadas he creído conveniente refactorizar el código para mayor claridad, detectar redundancias, simplificar su codificación y comentar con más detalle. No voy a entrar en los detalles, el resultado queda reflejado en la implementación y codificación final de las clases.

4.4.2.2.5 E.T 1.3 Añadir propiedades: Es plotter y tipo de papel.

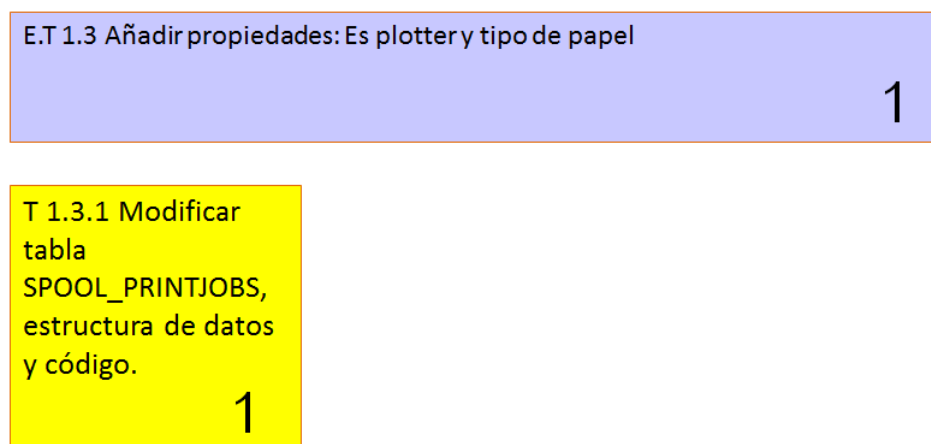


Ilustración 59. Tareas de E.T 1.3

T 1.3.1: Se modifica la tabla SPOOL_PRINTJOBS para añadir los campos ID_ISPLOTTER, ID_MEDIATYPE, N_MEDIATYPE. Además se han tenido que revisar las estructuras de datos y codificación del Monitor de impresión.

Revisando las fuentes de obtención de información (System.Printing, API y ficheros de spool), por un lado referente a los plotter, no se recupera ninguna propiedad en la que se haga referencia a si el dispositivo de impresión es un plotter; por otro lado, referente al tipo de papel utilizado, los dispositivos de impresión no detectan de forma automática el tipo de papel, se obtiene el tipo de papel que tiene configurado la impresora.

ID_ISPLOTTER Varchar2(1) // El dispositivo de impresión es un plotter S/N
 ID_MEDIATYPE Varchar2(256) // Descripción tipo de papel
 N_MEDIATYPE Number(5,0) // Código Tipo de Papel

Tabla 38. Nuevos campos para tabla SPOOL_PRINTJOBS para plotter y tipo de papel.

4.4.2.2.6 E.T 2.3 Revisión resultados obtenidos.

E.T 2.3 Revisión de resultados obtenidos

12

T 2.3.1 Cotejar información apuntada por usuarios y operadores con BD.

12

Ilustración 60. Tareas de E.T 2.3

T 2.3.1: El objetivo de esta tarea es revisar la información de los trabajos de impresión que se ha almacenado en BD teniendo en cuenta las anotaciones que han hecho los operadores a través del formulario Access y los usuarios en las hojas en papel.

4.4.2.2.7 E.T 2.2 Webform para que los operadores revisen los trabajos de impresión.

Este elemento de trabajo se descarta. Por que considero que el elemento de trabajo E.T 2.1 cubre los mismos objetivos.

4.4.2.3 Sprint Review (Reunión demostración)

Se muestra un resumen de los trabajos de impresión Capturados.

Durante los días que dura la iteración se capturan 100 trabajos de impresión. De todas las propiedades de los trabajos de impresión la única que presenta incidencias es el tamaño de papel en el plotter, al menos en el plotter “HP Designjet Z3200ps 44in Photo” para el que no se captura el tamaño del papel a través de las API del sistema de impresión de Windows. A continuación se presenta un resumen de los trabajos de impresión almacenados en BD:

Número total de impresiones por plotter:

PLOTTER	TRABAJO
HP 800. PCLXL	80
HP Z3200. EMF	20

Tabla 39. Total impresiones por plotter primera iteración..

Número total de copias impresas por documento:

PLOTTER	TRABAJO	COPIAS
HP 800. PCLXL	60	1
HP 800. PCLXL	2	10
HP Z3200. EMF	20	1

Tabla 40. Total impresiones por plotter según número de copias primera iteración.

Número total de impresiones por día:

PLOTTER	FECHA	TOTAL IMPRESIONES POR DIA
HP 800. PCLXL	06/04/2016	8
HP 800. PCLXL	07/04/2016	4
HP 800. PCLXL	08/04/2016	2
HP 800. PCLXL	11/04/2016	2
HP 800. PCLXL	12/04/2016	12
HP 800. PCLXL	13/04/2016	12
HP 800. PCLXL	15/04/2016	26
HP 800. PCLXL	18/04/2016	14
HP Z3200. EMF	12/04/2016	3
HP Z3200. EMF	13/04/2016	3
HP Z3200. EMF	15/04/2016	14

Tabla 41. Impresiones por día.

Número total de impresiones por tamaño de papel con comentario del operador:

PLOTTER	ANCHO	LARGO	REVISADO	NOTAS	TOTAL
HP 800. PCLXL	5940	8410			24
HP 800. PCLXL	6000	9000			12
HP 800. PCLXL	6000	9140			24
HP 800. PCLXL	9000	14200			2
HP 800. PCLXL	9140	11890			2
HP 800. PCLXL	9140	12000	x		2
				Ha sacado dos copias de este poster pero una de ellas se ha lanzado directamente desde el plotter con la opción reimprimir último trabajo.	
HP 800. PCLXL	9140	12000			1
HP 800. PCLXL	9140	12000		ok	1
HP 800. PCLXL	9140	12000			12
HP Z3200. EMF	0	0			20

Tabla 42. Impresiones por tamaño de papel.

4.4.2.4 Sprint retrospective

Los objetivos de esta iteración eran:

En primer lugar probar que se para cada trabajo de impresión capturado se obtiene toda la información requerida

Que el servicio monitor de impresión captura todos los trabajos de impresión y estos quedan registrados en BD.

En cuanto a los detalles de los trabajos de impresión capturados estos son los resultados:

	¿Se Captura?	Comentarios
Fecha	S	
Usuario	S	
Servidor	S	
Impresora	S	
Es plotter	N	El establecimiento de esta propiedad se incluirá dentro de la administración de dispositivos de impresión en el frontal web. También se puede obtener a partir del nombre del dispositivo de impresión si como parte del nombre se incluye el término "plotter".
Nombre de Documento	S	
Núm. total	S	Sólo tiene sentido para impresoras. Para plotter tiene sentido obtener el número de

páginas.		copias.
Num. Pág color	S	Tiene sentido para impresoras (Es necesario análisis de ficheros Spool). Para plotter consideramos que todas las impresiones son en color.
Tamaño papel	N	En plotter Z3200 no se obtiene a través de las API del sistema de impresión. Va a requerir obtener esta información analizando el fichero Spool (EMF).
Tipo Papel	N	Para el plotter HP 800 no se obtiene esta información y para el plotter HP Z3200 se obtiene el tipo de papel que tiene configurado el plotter esto no garantiza el tipo de papel utilizado.
Color	S	
Copias	S	

Tabla 43. Detalles trabajos de impresión iteración 1.

En cuanto al segundo punto, garantizar que se capturan todos los trabajos de impresión y quedan registrados en BD; con las anotaciones que hacen los usuarios en papel de los trabajos de impresión no podemos asegurar que todos los trabajos han sido registrados en BD, ya que los operadores tampoco han podido dedicar el tiempo necesario, por la saturación en su trabajo, para revisar los trabajos que se lanzaban por los plotter y hacer todas las anotaciones necesarias; como queda de manifiesto en la tabla resumen del punto anterior con el “número total de impresiones por tamaño con comentario del operador”. Por ello para la siguiente iteración tengo que establecer los mecanismos para poder cotejar fielmente los trabajos capturados en B.D con las anotaciones que hacen los usuarios en papel.

Además comentar que se abandonó la tarea para crear un WebForm en ASP.NET para que los operadores pudiesen incluir anotaciones en los trabajos de impresión registrados en BD, pues el formulario Access creado para este propósito cubría esta necesidad y así se pudo dedicar ese tiempo a la tarea E.T 2.3 (Revisión resultados obtenidos).

Para la siguiente Iteración se propone:

Añadir los siguientes elementos de trabajo a la funcionalidad F.2 del Product Backlog:

- Garantizar que se guarda en BD todos los trabajos de impresión y que los detalles obtenidos son correctos (E.T 2.4).
- Capturar el tamaño de página para el plotter HP Z3200 a través de fichero de spool. (E.T 2.5).
- Comprobar si el dispositivo es un plotter a través del nombre del dispositivo (E.T 2.6).

Para la siguiente iteración se va a hacer un esfuerzo y se va a aumentar la VT pasando de 50 horas por iteración a 70 horas, es decir 7 horas/día (de lunes a viernes).

4.4.3 Segunda Iteración

4.4.3.1 Sprint planning Meeting (Reunión de la iteración): Objetivos

Para esta iteración se pretende cubrir los nuevos objetivos identificados en la reunión retrospectiva de la iteración 1 añadiéndolos como nuevos elementos de trabajo a la funcionalidad F.2.

Además se incluye la nueva funcionalidad F.10 dedicada a la documentación del TFG que se va a realizar durante la segunda iteración y que es de obligado cumplimiento como parte de este TFG.

4.4.3.1.1 Elementos de trabajo de cada funcionalidad

4.4.3.1.1.1 F.2 Revisar y validar la información obtenido de los trabajos de impresión

La funcionalidad F.2 actualizada con los nuevos elementos de trabajo (no se incluyen los ya finalizados).

ID	Tarea	Estimación (Horas)
E.T 2.5	Garantizar calidad de los datos obtenidos	35
E.T 2.6	Captura tamaño de página en ficheros de Spool EMF	17
E.T 2.7	Comprobar si el dispositivo de impresión es un plotter	2

Tabla 44. nuevos E.T asociados a funcionalidad F.2

Product Backlog ID: F.10
Proyecto: Monitor Impresión

Prioridad
500

Estimación
16

Notas

Documentación del TFG.

Como probarlo

El tutor del TFG supervisa el progreso de la documentación del TFG y hace recomendaciones.

Ilustración 61. Funcionalidad F.10

Elementos de trabajo asociados a la funcionalidad F.10

ID	Tarea	Estimación (Horas)
E.T 10.1	Documentar. Introducción.	16

Tabla 45. . E.T asociados a funcionalidad F.10

4.4.3.2 Desarrollo iteración

La siguiente ilustración muestra la evolución de la iteración mediante una gráfica burndown

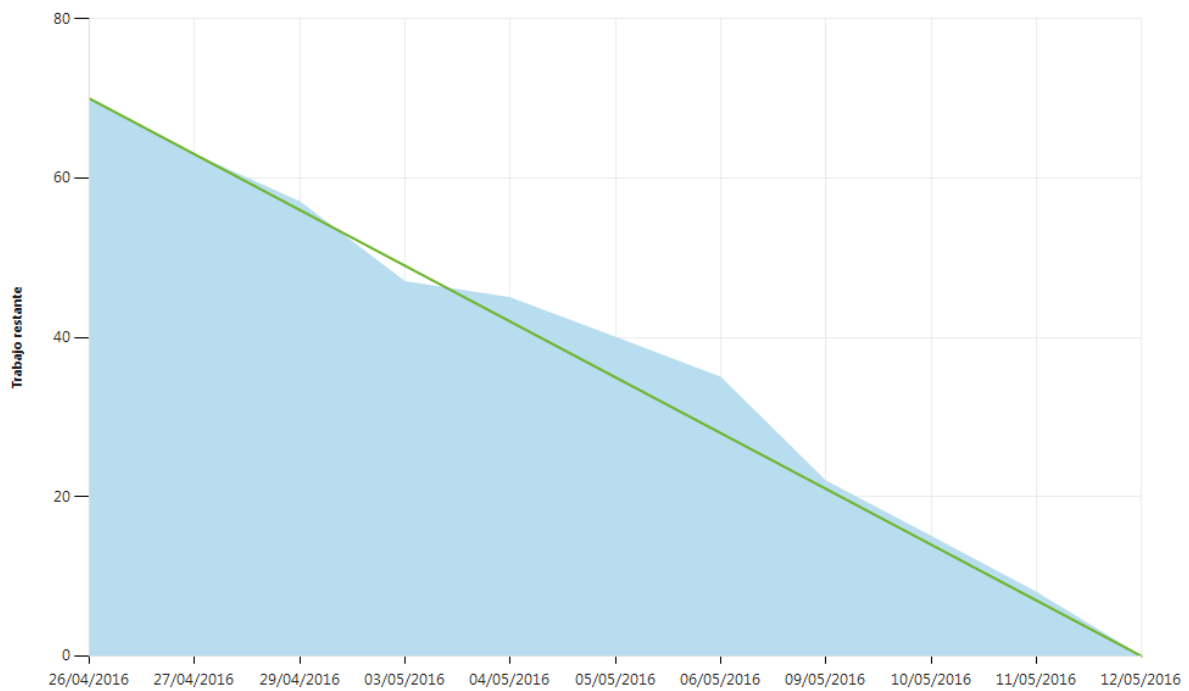


Ilustración 62. Gráfica burndown de la 2ª iteración.

4.4.3.2.1 E.T 2.5 Garantizar calidad de los datos obtenidos

E.T 2.5 Garantizar calidad de los trabajos obtenidos

35

T 2.5.1 Modificar hoja de registros de trabajos de impresión para usuarios

2

T 2.5.2 Implementar aviso por email de impresión en curso

5

T 2.5.3 Revisar in situ los trabajos de impresión y anotar en hoja de usuario

15

T 2.5.4 Traspaso de información de hojas de usuario a BD

6

T 2.5.5 Análisis de información capturada en BD. Contrastar con apuntes manuales


7

Ilustración 63. Tareas de E.T 2.5

El objetivo es por un lado asegurar que todos los trabajos de impresión se guardan en BD y por otro lado que las propiedades de impresión registradas son correctas tal como apuntan los usuarios en las hojas de registro manuales. Para ello se han establecido las siguientes tareas:

T 2.5.1: En primer lugar se requiere modificar la hoja de registros de trabajos de impresión para que se especifique toda la información relevante, que además nos permitirá identificar el trabajo en BD y verificar su corrección. La hoja se muestra en la siguiente ilustración e incluye la siguiente información para cada trabajo de impresión (se indican los nuevos campos):

- Fecha y hora de impresión (Campos nuevo)
- Apellidos y nombre
- Usuario (Campo nuevo)
- Tamaño del papel (Campo modificado)
- Número de copias
- Plotter por el que se imprime (Campo nuevo)
- Nombre del documento (Campo nuevo)
- Firma del usuario.



UNIVERSIDAD DE ALMERÍA

CONSUMO IMPRESIÓN

Centro de gastos: _____

Dpt/ Grupo de Investigación _____

Fecha	Hora(Aprox)	Apellidos, Nombre	Usuario	Tamaño(Alto, ancho)	Copias
Plotter (1)HP 800 (2)HP Z3200)		Nombre de Documento			Firma

Fecha	Hora(Aprox)	Apellidos, Nombre	Usuario	Tamaño (Alto, ancho)	Copias
Plotter (1)HP 800 (2)HP Z3200)		Nombre de Documento			Firma

Fecha	Hora(Aprox)	Apellidos, Nombre	Usuario	Tamaño(Alto, ancho)	Copias
Plotter (1)HP 800 (2)HP Z3200)		Nombre de Documento			Firma

MUY IMPORTANTE: Por favor rellenar todos los campos

Ilustración 64. Hoja de registro de trabajos de impresión.

En segundo lugar se modifica también la tabla SP00L_PRINTJOB para poder guardar en BD las anotaciones convenientes, junto a los datos apuntados por los usuarios en las hojas de registro de trabajos de impresión.

T 2.5.2: Se implementa el aviso por email cuando se lanza un nuevo trabajo de impresión a alguno de los dispositivos de impresión. Para implementarlo se utiliza Log4Net configurando un nuevo appender para envío de log a través de email y generando el log cuando se detecta el evento de impresión de documento.

En los anexos se dan más detalles sobre la librería Log4net.

En la siguiente ilustración se muestra un ejemplo de avisos recibidos por email:

```

cau17 -AnalyzeJobId- :: WARN ::
- ID_JOBNAME: 00010_13/05/2016_18:23:06
- ID_FUENTE: API
- N_JOB: 10
- F_PRINTJOB: 13/05/2016 18:23:06
- ID_LOGIN: usuario
- ID_PRINTSERVER: \\CAU17
- ID_PRINTER: Plotter HP Designjet 800PS 42 by HP
- ID_DOCUMENT: Poster B.I.R..pdf
- N_PAGES: 1
- N_PAGES_PRINTED: 1
- N_COLORPAGES: 1
- N_LENGTH: 10000
- N_WIDTH: 7070
- ID_MEDIASIZE:
- N_MEDIASIZE: 257
- ID_ORIENTATION: Portrait
- N_ORIENTATION: 1
- N_COPIES: 1
- ID_COLOR: Color
- N_COLOR: 2
- ID_DUPLEX: Simplex
- N_DUPLEX: 1
- ID_STATUS: 0
- ID_ISPLOTTER: S
- ID_MEDIATYPE:
- N_MEDIATYPE: 0
- ID_NOTAS:
- ID_REVISADO:
- N_JOBSIZE: 15804899
Logger: AnalyzeJobId
Thread: 6
Date: 2016-05-13 18:27:18,633
NDC: (null)

```

```

cau19 -AnalyzeJobId- :: WARN ::
- ID_JOBNAME: 00033_12/05/2016_11:46:08
- ID_FUENTE: EMF
- N_JOB: 33
- F_PRINTJOB: 12/05/2016 11:46:08
- ID_LOGIN: usuario
- ID_PRINTSERVER: \\CAU19
- ID_PRINTER: HP Designjet Z3200ps 44in Photo
- ID_DOCUMENT: lorcaimposiblecartelfinal.pdf
- N_PAGES: 1
- N_PAGES_PRINTED: 1
- N_COLORPAGES: 1
- N_LENGTH: 90400
- N_WIDTH: 117800
- ID_MEDIASIZE:
- N_MEDIASIZE: -1
- ID_ORIENTATION:
- N_ORIENTATION: -1
- N_COPIES: 0
- ID_COLOR: Color
- N_COLOR: -1
- ID_DUPLEX:
- N_DUPLEX: -1
- ID_STATUS:
- ID_ISPLOTTER: N
- ID_MEDIATYPE:
- N_MEDIATYPE: 0
- ID_NOTAS:
- ID_REVISADO:
- N_JOBSIZE: 394395648
Logger: AnalyzeJobId
Thread: 8
Date: 2016-05-12 11:46:50,086
NDC: (null)

```

Ilustración 65. Ejemplos de notificaciones por email de impresión en curso.

T 2.5.3: Esta tarea se va compaginando con el resto de tareas durante toda la iteración y consiste en que cuando se recibe una notificación por email de impresión en curso intento acercarme personalmente a la sala de usuarios del STIC para revisar in situ el trabajo de impresión que se ha lanzado y que sus detalles se apuntan correctamente en la hoja de registro de trabajos de impresión.

T 2.5.4: Con esta tarea se traspasan las anotaciones de los trabajos de impresión apuntada por los usuarios en las hojas de registro a los registros de BD que contienen información de los trabajos de impresión.

T 2.5.5: Con esta tarea se comprueba al final de la iteración si se han capturado todos los trabajos y si la información capturada es correcta. Para ello se lanzan consultas SQL a la BD y si es necesario se revisan logs.

4.4.3.2.2 E.T 2.6 Captura tamaño de página en ficheros de Spool EMF

E.T 2.6 Captura tamaño de página en ficheros de spool EMF

17

T 2.6.1 Revisión
estructuras EMF con
información de
tamaño de página

2

T 2.6.2 Análisis
muestreo ficheros
de spool EMF de
plotter HP Z3200

3

T 2.6.3 Análisis
estructura posibles
registros EMRI y
EMR para obtener
tamaño de página

2

T 2.6.4
Implementación
análisis EMF para
obtener tamaño de
página

5

T 2.6.5 Pruebas
implementación
tamaño página en
ficheros de spool
EMF

5

Ilustración 66. Tareas de E.T 2.6

El objetivo es obtener el tamaño de página del documento a través del análisis del fichero de spool EMFSPPOOL ya que para el plotter HP Z3200 que genera ficheros de spool en formato EMFSPPOOL no es posible obtener el tamaño de página a través de la API del sistema de Impresión. Para conseguirlo se han realizado las siguientes tareas:

T 2.6.1: A partir de los logs se revisan los resúmenes de tags utilizados en el análisis de cada fichero de Spool. De esta revisión se puede deducir que aparecen tantos registros EMR_HEADER como número de páginas tiene el trabajo de impresión guardado en el fichero de spool. El registro EMR_HEADER entre otras cosas guarda información del tamaño del área de trabajo utilizada en cada página del documento. Es necesaria haber completado anteriormente la tarea T 2.6.2.

T 2.6.2: Se debe obtener un resumen con los registros EMFSPPOOL y EMF (EMRI y EMR) utilizados, con totales por cada tipo de registro. Se crea la clase **Tags** que utilizamos en el análisis de ficheros de Spool para registrar los Tags del lenguaje utilizados así como el número de veces que se utiliza cada uno y la suma total de apariciones de tags.

Una muestra del resumen de tags en el análisis de un fichero de spool EMF:

```
.....
[05-05-2016 13:28:28,626] [INFO] -EmfSpool- Cerramos el fichero EMF
[05-05-2016 13:28:28,627] [INFO] -Items- -----
[05-05-2016 13:28:28,627] [INFO] -Items- Resumen Tags con el número de ocurrencias.
[05-05-2016 13:28:28,627] [INFO] -Items- -----
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_BEGINPATH: 8
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_CREATEBRUSHINDIRECT: 305858
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_DELETEOBJECT: 305858
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_ENDPATH: 8
```

```

[05-05-2016 13:28:28,627] [INFO] -Items- EMR_EOF: 1
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_FILLPATH: 8
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_HEADER: 1
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_INTERSECTCLIPRECT: 307161
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_MOVETOEX: 1
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_POLYDRAW16: 8
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_RECTANGLE: 7
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_RESTOREDC: 613290
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_SAVEDC: 613290
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_SELECTOBJECT: 611749
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_SETBRUSHORGEX: 2
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_SETCOLORSPACE: 1
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_SETICMMODE: 2
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_SETPOLYFILLMODE: 1
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_SETSTRETCHBLTMODE: 306071
[05-05-2016 13:28:28,627] [INFO] -Items- EMR_STRETCHDIBITS: 306071
[05-05-2016 13:28:28,627] [INFO] -Items- EMRI_METAFILE_DATA: 1
[05-05-2016 13:28:28,627] [INFO] -Items- EMRI_METAFILE_EXT: 1
[05-05-2016 13:28:28,627] [INFO] -Items- HEADER_RECORD: 1
[05-05-2016 13:28:28,627] [INFO] -Items- -----
[05-05-2016 13:28:28,627] [INFO] -Items- Total: 3369399
[05-05-2016 13:28:28,627] [INFO] -Items- -----
[05-05-2016 13:28:28,627] [INFO] -EmfSpool- FIN Analisis EMF
[05-05-2016 13:28:28,627] [INFO] -EmfSpool- -----

```

Detalle 43. Ejemplo de resumen de tags en análisis EMF

De los resúmenes se puede extraer información interesante por ejemplo el número de páginas que contiene el documento (EMRI_METAFILE_DATA) y que existen tantos registros de cabecera de página (EMR_HEADER) como número de páginas tiene el documento.

T 2.6.3: Hay 2 tipos de registros que guardan información del tamaño página:

- EMRI_DEVMODE: contiene una estructura DEVMODE con la configuración del dispositivo de impresión
- EMR_HEADER: Fichero cabecera con información de cada página. Contiene información del tamaño del área de impresión de la página.

Revisando los ficheros de logs y en particular el resumen de Tags empleados por cada trabajo de impresión se comprueba que siempre aparece un registro EMR_HEADER por cada página de los trabajos de impresión. Sin embargo no siempre aparece un registro EMRI_DEVMODE para un trabajo de impresión.

Por lo tanto se van a analizar los registros EMR_HEADER para obtener el tamaño de página. En los registros EMR_HEADER se define el tamaño del área de impresión de la página asociada.

T 2.6.4: Se implementa el análisis del registro EMR_HEADER para obtener el tamaño del área de impresión de la página asociada: método `AnalyzeEmrHeader()` de la clase `EmfSpool`.

T 2.6.5: Se lanzan las pruebas automáticas de análisis de ficheros de spool EMF para comprobar que no se produce ningún error. Se revisan los ficheros de log para comprobar que se está analizando el registro EMR_HEADER correctamente.

En el siguiente extracto del fichero de log generado al lanzar la prueba automática de análisis de ficheros de Spool Emf, se muestra como se obtiene correctamente el área (frame) o tamaño del recuadro que contiene la página definida por los dos puntos que conforman la diagonal del recuadro. Expresado en centésimas de milímetro.

```
>>>> Fichero de Spool para analisis EMF: D:\TestMonitorImpresion\PlotterEMF\00002.SPL
Análisis mediante lenguaje EMF del ficheros de Spool:
D:\TestMonitorImpresion\PlotterEMF\00002.SPL
Posición: (0x00000000) registro EMF: HEADER_RECORD Tamaño: 0x00000020.
Posicion siguiente registro EMFSPOOL: (0x00000020)
Posición: (0x00000020) registro EMF: EMRI_METAFILE_DATA Tamaño: 0x0000C7C8.
Posición 0x00000028 EMR_HEADER. Tamaño registro: 0x00000084
0x0000004C Frame: (0, 0), (21000, 29700)
Posición 0x000000AC EMR_SELECTOBJECT. Tamaño registro: 0x0000000C
```

Detalle 44. Análisis EMF. log ejemplo con obtención de tamaño de página

4.4.3.2.3 E.T 2.7 Comprobar si el dispositivo de impresión es un plotter

E.T 2.7 Comprobar si el dispositivo de impresión es un plotter

2

T 2.7.1
Implementación
para obtener la
propiedad si es
plotter

1

T 2.7.2 Prueba
obtener propiedad
si es plotter

1

Ilustración 67. Tareas de E.T 2.7

El objetivo es identificar si un dispositivo de impresión es un plotter comprobando si en el nombre está incluida la cadena plotter. Para que esto tenga sentido los dispositivos tipo plotter hay que identidicarlos al asignarles su nombre.

T 2.7.1: En la clase Util se implanta el método ContieneTexto() para comprobar si una cadena incluye una subcadena ignorando mayúsculas y mnúsculas. Este método se lanza cuando se recupera el nombre de la impresora y se actualizan las propiedades del trabajo de impresión.

T 2.7.2: Se crea una prueba unitaria para probar el método. Se comprueban en los logs que se obtiene correctamente la propiedad según el nombre del dispositivo de impresión.

4.4.3.2.4 E.T 10.1 Documentar. Introducción.

E.T 10.1 Documentar. Introducción.

16

T 10.1.1
Documentar TFG.
Introducción

16

Ilustración 68. . Tareas de E.T 10.1

T 10.1.1: Se documenta el Título de Introducción del TFG.

4.4.3.3 Sprint Review (Reunión demostración)

Se muestra un resumen de los trabajos de impresión Capturados.

Durante los días que dura la iteración se capturan 49 trabajos de impresión. De todas las propiedades de los trabajos de impresión la única que presenta incidencias es el tamaño de papel en el plotter, al menos en el plotter “HP Designjet Z3200ps 44in Photo” para el que no se captura el tamaño del papel a través de las API del sistema de impresión de Windows, pero esto se justifica en la reunión retrospectiva de la iteración. A continuación se presenta un resumen de los trabajos de impresión almacenados en BD:

Número total de impresiones por plotter:

PLOTTER	TRABAJOS
HP 800. PCLXL	27
HP Z3200. EMF	22

Tabla 46. Total impresiones por plotter segunda iteración.

Agrupación de trabajos según las copias que se han lanzado del mismo:

PLOTTER	TRABAJOS	COPIAS
HP 800. PCLXL	11	1
HP 800. PCLXL	1	2
HP 800. PCLXL	2	4
HP 800. PCLXL	1	6
HP Z3200. EMF	20	1
HP Z3200. EMF	1	2

Tabla 47. Total impresiones por plotter según número de copias en segunda iteración.

Número total de impresiones por día:

PLOTTER	FECHA	TOTAL IMPRESIONES POR DIA
HP 800. PCLXL	27/04/2016	2
HP 800. PCLXL	29/04/2016	16
HP 800. PCLXL	03/05/2016	2
HP 800. PCLXL	06/05/2016	2
HP 800. PCLXL	10/05/2016	1
HP 800. PCLXL	12/05/2016	4
HP Z3200. EMF	27/04/2016	4
HP Z3200. EMF	28/04/2016	1
HP Z3200. EMF	29/04/2016	9
HP Z3200. EMF	05/05/2016	4
HP Z3200. EMF	06/05/2016	3
HP Z3200. EMF	12/05/2016	1

Tabla 48. Número de impresiones por día segunda iteración.

Número total de impresiones por tamaño de papel con comentario del operador:

PLOTTER	ANCHO	LARGO	REVISADO	NOTAS	TOTAL
HP 800. PCLXL	4200	5940	S	OK. A2	12
HP 800. PCLXL	4200	5940	S	OK. A2. Cancelado por el usuario.	4
HP 800. PCLXL	8000	11600	S	OK. 11600x8000	1
HP 800. PCLXL	9140	12000	S	OK. A0	6
HP 800. PCLXL	9140	12000	S	OK. A0. Fallo Adobe.	2
HP 800. PCLXL	9140	12000	S	OK. Cancelado por el usuario	1
HP 800. PCLXL	9140	12190	S	OK. 12190x9140	1
HP Z3200. EMF	0	0	N	KO. 57800x90400. Sin apuntar.	1
HP Z3200. EMF	0	0	S	KO. 59000x89200. Cancelada impresión	1
HP Z3200. EMF	0	0	S	KO. 89200x59000.	1
HP Z3200. EMF	0	0	S	KO. 90400x117800	1
				KO. A2. OJO. Se lanzan 2 copias que es lo que se recoge a través del servicio Monitor Impresión. Sin embargo el plotter solo imprime una copia.	
HP Z3200. EMF	0	0	S	Tamaño papel A2. 39800x58400	2
HP Z3200. EMF	0	0	S	KO. A2. 58400x39800	1
HP Z3200. EMF	0	0	S	KO. Enviado por usuario. 147800x49000	1
HP Z3200. EMF	89000	87800	S	OK. 90x90	2
				KO. A0. El análisis en \\CAU19 ha fallado. Fichero grande, análisis sin completar fichero. Después se ha lanzado un análisis con un test de prueba y se ha analizado correctamente por que el fichero ya estaba completado.	
HP Z3200. EMF	117800	90400	S		1
HP Z3200. EMF	117800	90400	S	OK. A0	7
HP Z3200. EMF	269000	108800	S	OK. Cancelado usuario	2
HP Z3200. EMF	269000	110200	S	OK. Cancelado usuario	1
HP Z3200. EMF	270000	111000	S	OK. Cancelado usuario	1

Tabla 49. Impresiones por tamaño de papel segunda iteración.

4.4.3.4 Sprint retrospective

Los objetivos de esta iteración eran:

1. Poder probar definitivamente que para cada trabajo de impresión capturado se obtiene toda la información requerida y que se capturan todos los trabajos de impresión y quedan registrados en BD.
2. Después de la primera iteración el único detalle que no se capturaba era el tamaño de papel para el plotter HP Z3200, de aquí la necesidad de revisar el análisis de ficheros de spool EMF para obtener el tamaño de página.
3. Además se implementa una función para obtener a través del nombre del dispositivo si este es un plotter; esto no tiene ninguna dificultad no se hace falta comentar nada más respecto a este punto.

Respecto al primer punto, durante la primera iteración aparecen registrados 49 trabajos en BD se los que 48 se han apuntado en la hoja de registro, mientras que sólo uno no se ha apuntado en la hoja de registro y no se ha podido revisar (como se puede observar en la tabla “Impresiones por tamaño de papel segunda iteración”) por falta de disponibilidad para poder supervisarlo. Pero con estos datos podemos afirmar que el 100% de los trabajos se registran en BD, aunque haya usuarios que no hayan apuntado la información del trabajo de impresión en la hoja de registro.

En cuanto al segundo punto vemos que aparecen varios trabajos lanzados por el plotter HP Z3200 sin información del tamaño de la hoja de impresión. Esto se debe a que hasta el día 29 (tercer día de la segunda iteración) no se terminó la implementación del análisis de ficheros de spool EMF, sin embargo posteriormente se lanzó un test sobre los ficheros de spool correspondientes a estos trabajos de impresión y se obtuvo el tamaño de página correctamente como estaba apuntado en la hoja de registros de trabajos de impresión.

Por último se detecta un error en el análisis de un trabajo de impresión a través del plotter HP Z3200 donde sólo aparece la información del trabajo de impresión obtenida a través de la API del sistema de impresión, pero no aparece información obtenida a través del análisis EMF. Revisando los logs encuentro la siguiente información en relación con el error:

```
.....
[05-05-2016 12:56:26,710] [INFO] -WatchIO- Creación fichero:
C:\Windows\System32\spool\PRINTERS\00027.SPL
.....
[05-05-2016 12:56:30,697] [INFO] -EmfSpool- >>>> Fichero de Spool para analisis EMF:
C:\Windows\System32\spool\PRINTERS\00027.SPL

[05-05-2016 12:56:30,898] [INFO] -EmfSpool- Tamaño de fichero SPL para análisis EMF:
654550416

[05-05-2016 12:56:30,899] [ERROR] -EmfSpool- No se puede abrir el fichero de spool:
C:\Windows\System32\spool\PRINTERS\00027.SPL

[05-05-2016 12:56:30,902] [FATAL] -AnalyzeJobId- System.IO.IOException: El proceso no
puede obtener acceso al archivo 'C:\Windows\System32\spool\PRINTERS\00027.SPL' porque
está siendo utilizado en otro proceso.
    en MonitorTrabajosImpresion.EmfSpool.AnalyzeEmf(STRUCT_PRINT_JOB& PrintJob)
    en MonitorTrabajosImpresion.AnalyzeJobId.AnalyzeSpool(String Printer, String JobId)
.....
```

Detalle 45. Error análisis fichero spool EMF. Fichero en uso

Vemos que entre el momento en el que se detecta la creación del fichero ([05-05-2016 12:56:26,710]) y el momento en el que se intenta abrir para su análisis ([05-05-2016 12:56:30,899]) transcurren 4 segundos aproximadamente. El fichero tiene un tamaño de 624 MB aproximadamente (654550416 bytes). Esto parece indicar por el mensaje [FATAL] devuelto por el sistema que el fichero todavía estaba siendo procesado por el Sistema de Impresión.

Posteriormente se ha lanzado una prueba de análisis EMF sobre el fichero de spool, ya había sido liberado, y se ha obtenido el tamaño de la página sin problema.

Para la siguiente Iteración se propone:

Añadir el siguiente elemento a la funcionalidad F.2 del Product Backlog:

- Corregir error en análisis de ficheros de Spool de gran tamaño (E.T 2.8).

Para la siguiente iteración se vuelve a reducir la VT a 50 horas es decir 5 horas/día (de lunes a viernes).

4.4.4 Tercera iteración

4.4.4.1 Sprint planning Meeting (Reunión de la iteración): Objetivos

Para esta iteración se pretende resolver el problema detectado con el análisis de ficheros de spool, según parece con ficheros de spool de gran tamaño. Para ello se añade un nuevo elemento de trabajo a la funcionalidad F.2 del Product Backlog.

Además se sigue con la funcionalidad F.10 dedicada a la documentación del TFG que se va a realizar durante la tercera iteración de obligado cumplimiento como parte de este TFG.

4.4.4.1.1 Elementos de trabajo de cada funcionalidad

4.4.4.1.1.1 F.2 Revisar y validar la información obtenida de los trabajos de impresión

La funcionalidad F.2 actualizada con el nuevo elemento de trabajo (no se incluyen los ya finalizados).

ID	Tarea	Estimación (Horas)
E.T 2.8	Corregir error en análisis de ficheros de Spool de gran tamaño	15

Tabla 50. nuevo E.T asociado a funcionalidad F.2

4.4.4.1.1.2 F.10 Documentar TFG

La funcionalidad F.10 actualizada con el nuevo elemento de trabajo (no se incluye el ya finalizado).

ID	Tarea	Estimación (Horas)
E.T 10.2	Documentar. Estado del arte.	35

Tabla 51. nuevo E.T asociado a funcionalidad F.10

4.4.4.2 Desarrollo iteración

La siguiente ilustración muestra la evolución de la iteración mediante una gráfica burndown

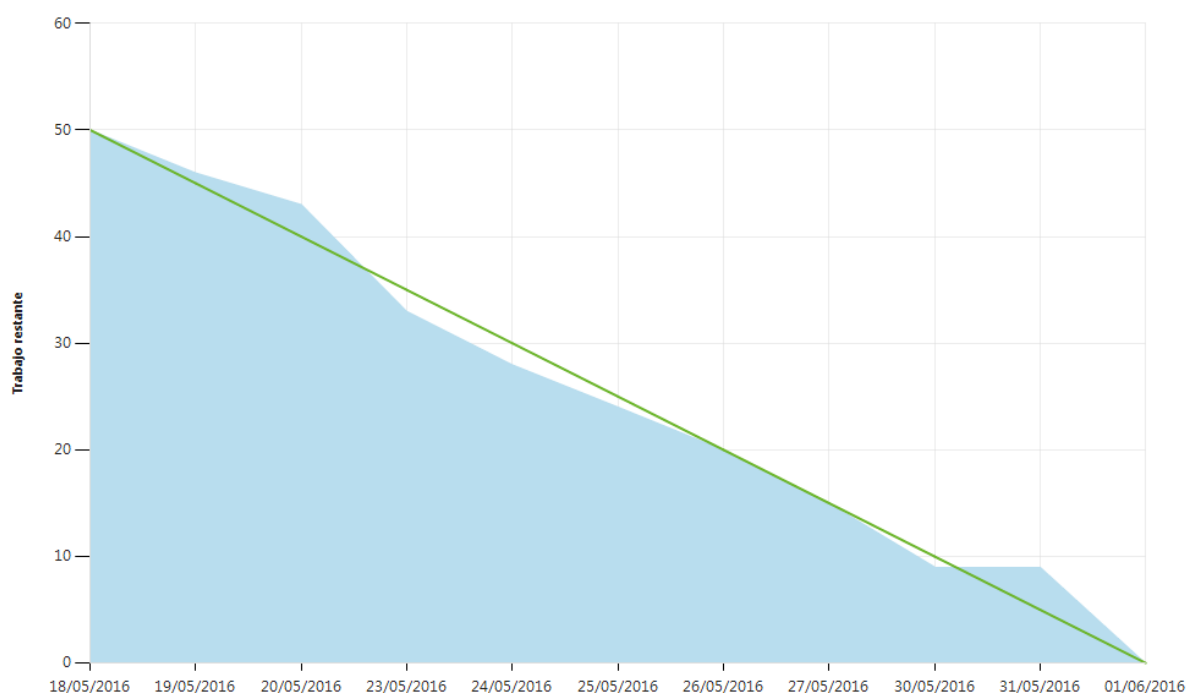


Ilustración 69. Gráfica burndown de la 3ª iteración

4.4.4.2.1 E.T 2.8 Corregir error en análisis de ficheros de Spool de gran tamaño

E.T 2.8 Corregir error en análisis de ficheros de Spool de gran tamaño

15

T 2.8.1 Investigar cual es la posible causa del error en análisis del fichero de spool EMF.

2

T 2.8.2 Implementar una simulación para reproducir la causa del error.

8

T 2.8.3 Refactorizar análisis de ficheros de spool para corregir el bug detectado.

5

Ilustración 70. Tareas de E.T 2.8

El objetivo es descubrir por que se ha producido el error al analizar un fichero de spool EMF de un trabajo de impresión, reproducir el error y por último hacer las correcciones necesarias para que no se vuelva a producir el error.

T 2.8.1: Tras revisar los ficheros de log con más detalle no descubro nada nuevo aparte de lo comentado en la reunión retrospectiva de la segunda iteración. Todo señala a que cuando el sistema de impresión genera ficheros de spool con trabajos de impresión demasiado grandes el Monitor de Impresión intenta realizar el análisis del fichero de spool y este todavía sigue siendo utilizado (posiblemente no se haya acabado de crear) por el sistema de impresión.

T 2.8.2: La implementación de la simulación la he ideado utilizando la siguiente prueba automatizada con Visual Studio:

Para la prueba se copian en un directorio varios ficheros de spool EMF de gran tamaño entre ellos con el que se detecto el error.

Para cada ArchivoOriginal contenido en el directorio:

1. Se crea un thread para hacer una copia del ArchivoOriginal en un ArchivoBackup.
2. En el hilo principal se intenta abrir el ArchivoBackup en modo exclusivo. Si da un error se repite la operación durante un tiempo límite que establecemos nosotros.
3. Se comprueba si se ha podido abrir el ArchivoBackup.

Detalle 46. T 2.8.2 lógica de prueba copia ficheros grandes

Para esta prueba jugando con el tiempo que se pone como límite para poder abrir el archivo de backup en modo exclusivo, el test se ejecuta correctamente o sin éxito.

He probado poniendo como tiempo límite 4 segundos, que era aproximadamente el tiempo que aparecía en los logs desde que se crea el fichero hasta que se intenta realizar el análisis del fichero. El test falla.

También falla estableciendo el tiempo límite en 10 y 15 segundos.

Poniendo el tiempo límite a 20 segundos el test se ejecuta correctamente.

Así que los resultados del test parecen razonables y tiene sentido pensar que el error se produce cuando se generan ficheros de spool grandes.

El test creado, "`public void WatchIO_IsFileLocked_ComprobarSiUnFicheroEstaBloqueado()`", se incluye como parte del código. Ver anexo.

T 2.8.3: Teniendo en cuenta el resultado de la prueba realizada se hacen las correcciones necesarias en la codificación del Servicio Monitor de Impresión de manera que antes de realizar el análisis de los ficheros de spool se chequea que el fichero no está bloqueado.

4.4.4.2.2 E.T 10.2 Documentar. Estado del arte.

E.T 10.2 Documentar. Estado del arte.

35

T 10.2.1
Documentar TFG.
Estado del arte

35

Ilustración 71. Tareas de E.T 10.2

Se documenta el título dedicado al estado del arte del TFG.

4.4.4.3 Sprint Review (Reunión demostración)

Se explican las evidencias encontradas relativas al error producido al realizar el análisis del fichero de spool del trabajo de impresión lanzado por el plotter HP Z3200 y las medidas que he

tomado para corregirlo. Aclarando que este error se podía haber producido para cualquier dispositivo de impresión que requiera análisis de los ficheros de spool para obtener algún detalle de los trabajos de impresión.

4.4.4.4 *Sprint retrospective*

Referente al trabajo realizado hasta el momento todos nos damos por satisfechos.

La continuación del trabajo para calcular costes de impresión, es decir el frontal para explotar la información capturada por el servicio monitor de impresión, queda pospuesta hasta que finalice el TFG.

En cuanto a la fase SCRUM también la doy por finalizada para este TFG pues ya ha cumplido con el objetivo para el que la necesitaba. A partir de este punto sólo me queda concluir la documentación del TFG.

4.5 Valoración Económica

El costo total del proyecto teniendo en cuenta la valoración de recursos humanos y materiales (SW y HW):

Nombre	Costo
TFG Monitorizar Impresión Windows	12.691,60 €

Tabla 52. Coste total del TFG.

El coste se ha calculado en base a los costes por recursos humanos y recursos materiales como se muestra en los siguientes apartados.

4.5.1 Recursos humanos

La siguiente tabla muestra el coste por recurso humano/hora.

Nombre	Precio hora
Coordinador	20,00 €/hora
Desarrollador	10,00 €/hora
Desarrollador_10%	10,00 €/hora
Desarrollador_15%	10,00 €/hora
Desarrollador_25%	10,00 €/hora
Desarrollador_50%	10,00 €/hora
Desarrollador_62,5%	10,00 €/hora
Desarrollador_87,5%	10,00 €/hora
ProductOwner	10,00 €/hora
ScrumMaster	15,00 €/hora
ScrumMaster_5%	15,00 €/hora

Tabla 53. Coste de recurso humano por hora.

La siguiente ilustración muestra la asignación que se ha hecho de recursos humanos a tareas:

Nombre de tarea	Duración	Nombres de los recursos
TFG Monitorizar Impresión Windows	288 días	Coordinador[5%];Portatil[1]
PRIMERA FASE: Viabilidad (Prototipos)	77 días	
Requerimientos y Diseño inicial	5 días	Desarrollador_50%[50%];ProductOwner[50%]
Estudio Sistema Impresión Windows. Búsqueda información	72 días	Desarrollador_15%[15%]
Herramientas desarrollo. Formación.	72 días	Desarrollador_10%[10%]
Probar métodos captura trabajos de impresión	72 días	Desarrollador_25%[25%]
SEGUNDA FASE: Implementación (Modelo incremental)	53 días	
Requerimientos y Diseño	4 días	Desarrollador_62,5%[63%];ProductOwner[63%]
Capturar Eventos Impresión	12 días	Desarrollador_62,5%[63%]
Obtener trabajos de impresión a través de API	12 días	Desarrollador_62,5%[63%]
Obtener trabajos de impresión a través de Spool	25 días	Desarrollador_62,5%[63%]
TERCERA FASE: Implantación y pruebas (Scrum)	48 días	
Formación SCRUM, Visual Studio	8 días	
Reunión planificación: Product Backlog	3 horas	Desarrollador;ProductOwner;ScrumMaster
Sprint Planning Meeting Iteración 1	2 horas	Desarrollador;ScrumMaster
Iteración1	10 días	
Detectar plotter, tipo papel.Refactorización y pruebas automáticas.	7 días	Desarrollador_62,5%[63%];ScrumMaster_5%[5%]
Instalar Servicio Monitor: revisar trabajos de impresión obtenidos	3 días	Desarrollador_62,5%[63%];ScrumMaster_5%[5%]
Sprint review de la iteración 1	2 horas	Desarrollador;ProductOwner;ScrumMaster
Sprint retrospectiva de la iteración 1	2 horas	Desarrollador;ProductOwner;ScrumMaster
Sprint Planning Meeting Iteración 2	4 horas	Desarrollador;ScrumMaster
Iteración2	10 días	
Garantizar calidad en captura Trabajos de impresión	5 días	Desarrollador_87,5%[88%];ScrumMaster_5%[5%]
Corregir captura tamaño página en Spool EMF. Pruebas.	3 días	Desarrollador_87,5%[88%];ScrumMaster_5%[5%]
Documentar TFG	2 días	Desarrollador_87,5%[88%]
Sprint review de la iteración 2	1 hora	Desarrollador;ProductOwner;ScrumMaster
Sprint retrospectiva de la iteración 2	1 hora	Desarrollador;ProductOwner;ScrumMaster
Sprint Planning Meeting Iteración 3	4 horas	Desarrollador;ScrumMaster
Iteración3	10 días	
Corregir problemas de acceso a ficheros de Spool	3 días	Desarrollador_62,5%[63%];ScrumMaster_5%[5%]
Documentar TFG	7 días	Desarrollador_62,5%[63%]
Sprint review de la iteración 3	1 hora	Desarrollador;ProductOwner;ScrumMaster
Sprint retrospectiva iteración 3	2 horas	Desarrollador;ProductOwner;ScrumMaster
Documentar TFG	18 días	Desarrollador_87,5%[88%]

Ilustración 72. Recursos humanos asignados a tareas.

La siguiente tabla muestra un resumen del costo por cada recurso humano:

Nombre	Comienzo	Fin	Trabajo	Costo
Desarrollador_10%	vie 22/05/15	lun 31/08/15	57,6 horas	576,00 €
Desarrollador_15%	vie 22/05/15	lun 31/08/15	86,4 horas	864,00 €
Desarrollador_25%	vie 22/05/15	lun 31/08/15	144 horas	1.440,00 €
Desarrollador_50%	vie 15/05/15	jue 21/05/15	20 horas	200,00 €

Desarrollador_62,5%	lun 11/01/16	mar 31/05/16	365 horas	3.650,00 €
Desarrollador_87,5%	lun 25/04/16	vie 24/06/16	196 horas	1.960,00 €
Desarrollador	mar 05/04/16	mié 01/06/16	22 horas	220,00 €
Coordinador	vie 15/05/15	vie 24/06/16	115,2 horas	2.304,00 €
ScrumMaster	mar 05/04/16	mié 01/06/16	22 horas	330,00 €
ScrumMaster_5%	mié 06/04/16	vie 20/05/16	8,4 horas	126,00 €
ProductOwner	vie 15/05/15	mié 01/06/16	52,17 horas	521,60 €

Tabla 54. Coste total por recurso humano.

En algunos casos para los mismos recursos humanos se han hecho varias distinciones según su dedicación a las distintas tareas del proyecto. Por ejemplo se puede observar en el periodo del 22/05/2015 al 31/08/2015 que el desarrollador aparece desglosado en porcentajes del 10%, 15% y 25%, esto es así para indicar que trabaja simultáneamente en varias tareas dedicándole a cada una de ellas el % de tiempo indicado.

4.5.2 Costo por recursos materiales

Recurso material	Comienzo	Fin	Costo
Visual Studio Enterprise 2015	jue 21/05/15	vie 24/06/16	0,00 €
Hex Editor XVI32	vie 15/05/15	vie 24/06/16	0,00 €
Portatil	jue 21/05/15	vie 24/06/16	500,00 €
VisualStudio Team Foundation Server 2015	jue 24/03/16	mié 01/06/16	0,00 €
Visor de Eventos de Windows	jue 21/05/15	vie 24/06/16	0,00 €
Toad for Oracle 12.5	lun 11/01/16	mié 01/06/16	0,00 €
Toad Data Modeler 5.2	lun 11/01/16	mié 01/06/16	0,00 €
Notepad++	jue 21/05/15	vie 24/06/16	0,00 €
PdfCreator	vie 15/05/15	vie 24/06/16	0,00 €
Doxygen	jue 21/05/15	vie 24/06/16	0,00 €

Microsoft Project 2013	jue 21/05/15	vie 24/06/16	0,00 €
Office 2010	jue 21/05/15	vie 24/06/16	0,00 €
SGBD Oracle	lun 11/01/16	mié 01/06/16	0,00 €
IIS 7.5	lun 11/01/16	mié 23/03/16	0,00 €
HP LaserJet 5500	lun 11/01/16	mié 23/03/16	0,00 €
Plotter HP Designjet 800PS	jue 24/03/16	mié 01/06/16	0,00 €
HP Designjet Z3200ps	jue 24/03/16	mié 01/06/16	0,00 €
Administración Impresión Microsoft	jue 21/05/15	vie 24/06/16	0,00 €
Consumibles	jue 21/05/15	vie 24/06/16	20,00 €

Tabla 55. Costo por recursos materiales.

Gracias al acuerdo de la Universidad con Microsoft (programa DreamSpark) no se ha tenido que hacer ninguna inversión para poder utilizar los productos de Microsoft. Tampoco para el esto de productos SW que se han utilizado bajo alguno de los tipos de licencia gratuita. De igual manera lo mismo ha sucedido con los dispositivos de impresión y equipos utilizados; se han aprovechado los ya existentes sin ninguna repercusión, ni coste adicional salvo en consumibles (papel, toner), y solamente para las pruebas realizadas por impresoras por que las pruebas de impresión realizadas por los plotter se han realizado aprovechando los trabajos de impresión lanzados por los usuarios.

Por tanto los únicos costes materiales como consecuencia del TFG han sido: un ordenador portátil y consumibles (papel y toner).

5. CONCLUSIONES Y TRABAJO FUTURO

Cuando comencé este TFG no conocía las dificultades que me iba a encontrar y tampoco el esfuerzo y por supuesto el tiempo que iba a necesitar. Sabía por donde tenía que empezar y también sabía cual debía ser el resultado final. Por donde empezar no podía ser de otra manera que entender suficientemente el funcionamiento del Sistema de Impresión Windows para descubrir la forma de interactuar con él; eso llevaba aparejado el aprendizaje de las herramientas necesarias para lograr esa interacción. Creo que ha sido una buena decisión utilizar Visual Studio y C# como lenguaje de programación, me ha facilitado mucho el trabajo.

El resultado final ha sido la creación de un Servicio Windows con el que es posible monitorizar los trabajos de impresión que los usuarios lanzan por los dispositivos de impresión y obtener sus características. Esta información se guarda en BD, por lo que también ha sido necesario crear un servicio WCF que implementa la capa de acceso a datos.

Entre medio ha habido momentos de optimismo y momentos en los que parecía retroceder más que avanzar, pero es así como se aprende y esa creo que es la mejor parte, los conocimientos adquiridos: Familiarizarme con el entorno Visual Studio, lenguaje C#, interactuar con el Servicio de gestión de Eventos, aprender a utilizar las API del Servidor de Impresión, comprender los lenguajes PDL utilizados (EMF, EMFSPool, JPL y PCLXL hasta el punto de ser capaz abrir los ficheros de spool y pelearme con su contenido en formato hexadecimal), ser capaz de crear un servicio Windows y su instalador, también crear un servicio WCF y ofrecerlo a través de un servidor IIS para implementar la capa de acceso a datos, descubrir las metodologías ágiles en concreto Scrum y una herramienta que le de soporte en mi caso Visual Studio TFS. La lista no acaba aquí pero creo que los ejemplos dados son una muestra suficiente para ver que ha merecido la pena.

En cuanto al alcance del trabajo. Se han realizado pruebas en distintas impresoras, además de la ofrecida por el servicio de impresión del STIC, pero para probar el funcionamiento real del Servicio Monitor de Impresión se ha comenzado por los plotter ofrecidos por el servicio de impresión del STIC, por dos razones; la primera por que no interfiere con el funcionamiento normal de la aplicación Megatrack con la que ya se están calculando costes de impresión de forma automatizada (al menos en parte) y la segunda por que parece razonable dar el siguiente paso consistente en automatizar el cálculo de costes de impresión precisamente en estos dispositivos, plotters, para los que todavía se hace de forma manual mientras que para la impresora se pueden seguir obteniendo, de momento, con la aplicación Megatrack.

¿Qué podrá encontrar el lector en este TFG?, las claves para interactuar con el Registro de Eventos de Windows y conocer cuales son los eventos asociados al Sistema de impresión más importantes. También se dan las claves para utilizar las API del Sistema de Impresión y las estructuras necesarias para poder obtener las propiedades de los trabajos de impresión. Por último se profundiza en la comprensión de varios lenguajes PDL: EMF, EMFSPool, JPL y PCLXL

utilizados en los ficheros de spool para también poder obtener propiedades de los trabajos de impresión.

El trabajo inmediatamente futuro es desarrollar una aplicación (frontal) que permita recoger los frutos de este trabajo y poder calcular los costes de los trabajos de impresión capturados; primero para los plotters y posteriormente para impresoras, pudiendo reemplazar el SW Megatrack.

Además para futuras adquisiciones de dispositivos de impresión es muy probable que haya que implementar el análisis de ficheros de spool para otros lenguajes PDL como XPS o PostScript.

También habrá que revisar la seguridad en las comunicaciones entre los componentes que conformarán la aplicación, en especial entre los módulos ya implementados: Servicio Monitor de Impresión y el servicio WCF que ofrece la capa CAD.

6. BIBLIOGRAFÍA

- [MS-EMF]: Enhanced Metafile Format. (n.d.). Retrieved May 11, 2016, from <https://msdn.microsoft.com/en-us/library/cc230514.aspx>
- [MS-EMFPLUS]: Enhanced Metafile Format Plus Extensions. (n.d.). Retrieved May 11, 2016, from <https://msdn.microsoft.com/en-us/library/cc230724.aspx>
- [MS-RPRN]: Print System Remote Protocol. (n.d.). Retrieved May 19, 2016, from <https://msdn.microsoft.com/en-us/library/cc244528.aspx>
- Alvorsen, H.-P. H. (2012). Introduction to Visual Studio and C #, 1–48. Retrieved from [http://home.hit.no/~hansha/documents/microsoft.net/tutorials/introduction to visual studio/Introduction to Visual Studio and CSharp.pdf](http://home.hit.no/~hansha/documents/microsoft.net/tutorials/introduction%20to%20visual%20studio/Introduction%20to%20Visual%20Studio%20and%20CSharp.pdf)
- Apache Log4Net. (n.d.). Log4Net. Retrieved June 24, 2016, from <https://logging.apache.org/log4net/>
- Cadavid, A. N., Fernández Martínez, J. D., & Morales Vélez, J. (2013). Revisión de metodologías ágiles para el desarrollo de software A review of agile methodologies for software development. *Universidad Icesi*, 11 No. 2, 30–39. Retrieved from dialnet.unirioja.es/descarga/articulo/4752083.pdf
- Corey, T. (2015). Log4net Tutorial. Retrieved June 25, 2016, from <http://www.codeproject.com/Articles/140911/log4net-Tutorial>
- Ferguson, J., Patterson, B., Beres, J., Boutquin, P., & Gupta, M. (n.d.). *C # Bible*.
- Figuerola, N. (2004). Kanban, Su Uso en el Desarrollo de Software. *Journal of Personality*, 8.
- Harold, E. R., & Means, W. S. (2005). *XML*. ANAYA MULTIMEDIA. Retrieved from <https://books.google.es/books?id=xPokAAAACAAJ>
- Inteco, E. (Instituto N. de T. de la C. (2009). Ingeniería del Software: Metodologías y Ciclos de Vida. *Inteco*, 83. Retrieved from https://www.incibe.es/file/N85W1ZWfHifRgUc_oY8_Xg
- Kniberg, H. (2007). *Scrum and XP desde las trincheras*. Reading. <http://doi.org/10.1017/S1474745605002673>
- Manifiesto por el Desarrollo Ágil de Software. (n.d.). Retrieved March 12, 2016, from <http://agilemanifesto.org/iso/es/manifesto.html>
- MSDN de Microsoft. (n.d.). SetPrinter API function. Retrieved June 14, 2016, from [https://msdn.microsoft.com/en-us/library/windows/desktop/dd145082\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd145082(v=vs.85).aspx)
- MSDN Microsoft. (n.d.-a). ClosePrinter API Function. Retrieved June 14, 2016, from [https://msdn.microsoft.com/en-us/library/windows/desktop/dd183446\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd183446(v=vs.85).aspx)
- MSDN Microsoft. (n.d.-b). DEVMODE structure. Retrieved June 17, 2016, from [https://msdn.microsoft.com/en-us/library/windows/desktop/dd183565\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd183565(v=vs.85).aspx)
- MSDN Microsoft. (n.d.-c). Documents and Printing. Retrieved June 29, 2016, from [https://msdn.microsoft.com/en-us/library/windows/desktop/ff686798\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ff686798(v=vs.85).aspx)
- MSDN Microsoft. (n.d.-d). GetJob API function. Retrieved June 14, 2016, from [https://msdn.microsoft.com/en-us/library/windows/desktop/dd144894\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd144894(v=vs.85).aspx)
- MSDN Microsoft. (n.d.-e). GetPrinter API Function. Retrieved June 14, 2016, from [https://msdn.microsoft.com/en-us/library/windows/desktop/dd144911\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd144911(v=vs.85).aspx)
- MSDN Microsoft. (n.d.-f). JOB_INFO_2 structure. Retrieved June 14, 2016, from

- <https://technet.microsoft.com/es-es/library/dd145020.aspx>
- MSDN Microsoft. (n.d.-g). OpenPrinter API function. Retrieved June 16, 2016, from [https://msdn.microsoft.com/en-us/library/windows/desktop/dd162751\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd162751(v=vs.85).aspx)
- MSDN Microsoft. (n.d.-h). Print Spooler API Structures. Retrieved June 17, 2016, from [https://msdn.microsoft.com/en-us/library/windows/desktop/dd162864\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd162864(v=vs.85).aspx)
- MSDN Microsoft. (n.d.-i). System.Printing Namespace. Retrieved from [https://msdn.microsoft.com/en-us/library/system.printing\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.printing(v=vs.110).aspx)
- MuchoCodigo. (n.d.). Clase de conexión a Base de Datos Oracle con C# .NET. Retrieved June 17, 2016, from <http://muchocodigo.com/clase-de-conexion-base-de-datos-oracle-con-c-net/>
- Packard, H. (2002). PCL XL Feature Reference Protocol Class 3.0 Supplement PCL XL Feature Reference Protocol Class 3.0 Supplement Document Revision History.
- PCL XL Feature Reference Protocol Class 2.0 Revision: p2.2 Word for Windows File: xl_ref20r22.doc Revision Date: March16,2000 Word for Windows Version: Word97 Author(s): FIGURE FILES: VISIO 4.0. (n.d.).
- Pérez, D. A., Ginestá, M. G., Matías, M. H., & Hernández, J. M. (2007). Ingeniería del software en entornos de SL, 314.
- Pressman, R. S. (2002). *Ingeniería del software. Un enfoque práctico*. Ingeniator. <http://doi.org/10.1073/pnas.0703993104>
- Print Spooler API Functions (Windows). (n.d.). Retrieved May 4, 2016, from [https://msdn.microsoft.com/en-us/library/windows/desktop/dd162861\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd162861(v=vs.85).aspx)
- Reino Romero, A. (2011). Manual XML. Retrieved June 24, 2016, from <http://www.desarrolloweb.com/manuales/manual-introduccion-xml.html>
- Schwaber, K., & Sutherland, J. (2013). The Scrum Guide. *Scrum.Org and ScrumInc.* <http://doi.org/10.1053/j.jrn.2009.08.012>
- System.Printing (Espacio de nombres). (n.d.). Retrieved June 21, 2015, from [https://msdn.microsoft.com/es-es/library/system.printing\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/system.printing(v=vs.110).aspx)
- Types, D. (1990). Core C # and . NET Quick Reference, (120), 1-2.
- umeshb. (n.d.). Cambiar propiedades de Impresora. Retrieved June 14, 2016, from <http://www.codeproject.com/Articles/6899/Changing-printer-settings-using-C>
- Undocumented Printing Wiki [Undocumented Printing]. (n.d.). Retrieved June 29, 2016, from <http://www.undocprint.org/start>
- Windows Protocols. (n.d.). Retrieved May 7, 2016, from <https://msdn.microsoft.com/en-us/library/cc216517.aspx>

7. ■ ANEXOS

7.1 Creación de un Servicio WCF para la capa CAD.

Para ello se crea un proyecto tipo Biblioteca WCF (.NET 4.0).

El proyecto consta de un interface (con la que se define el contrato WDSL) y la clase que la implementa.

La configuración del servicio WCF se puede revisar editando la configuración del archivo App.Config con el editor de configuración WCF (pinchando con el botón derecho del ratón sobre el fichero App.Config). Entre otras cosas se puede editar el binding entre extremos, por ejemplo wsHttpBinding para cifrar el mensaje (basicHttpBinding comunicación sin cifrar).

Para poder generar logs se agrega la librería log4net a través del gestor de paquetes integrado en Visual Studio 15 NuGet.

Para poder interactuar con BD Oracle es necesario tener instalado el cliente Oracle 12g, con la extensión ODP.NET, y agregar la librería Oracle.DataAccess.dll (.net 4.0) al proyecto.

Una vez realizado el Servicio WCF se puede ofrecer a través de un servidor Web como IIS (7.x o superior). Habilitando en IIS el servidor Ftp se puede publicar directamente desde Visual Studio el servicio WCF dentro de un sitio de IIS.

Una vez que tenemos publicado nuestro servicio WCF en un sitio IIS, el directorio donde se encuentra la estructura de archivos del servicio WCF hay que convertirlo en aplicación.

Como parte de la configuración de IIS hay que tener en cuenta algunos detalles:

- Habilitar .NET en IIS: para ello nos situamos en el framework de .NET (4.x) que queremos habilitar, p.e, C:\Windows\Microsoft.NET\Framework(64)\v4.0.30319 y ejecutamos el siguiente comando en una ventana de línea de comandos: `>aspnet_regiis -ir`
- En el grupo de aplicaciones de la aplicación WCF habilitamos .NET.
- Hay que levantar el sitio y el grupo de aplicaciones con un usuario que tenga los permisos necesarios en el directorio del servicio WCF (en Windows 2008 el usuario IIS_IUSR).
- También se puede habilitar la exploración de directorios de la aplicación WCF.
- Para ver la URL que tenemos que utilizar en la aplicación .NET (C#) donde se va a agregar el servicio WCF como referencia, desde IIS podemos pinchar en el panel lateral de la derecha sobre "Examinar aplicación". En la página se da información de la URL de referencia así como la forma de utilizar el servicio WCF en una aplicación cliente.

En la aplicación cliente hay que agregar una referencia de servicio WCF, en la que se utiliza la URL que se obtiene al examinar aplicación WCF en IIS.

Una vez hecho esto para utilizar el servicio WCF hay que indicar la referencia al servicio agregado con una clausula using.

7.2 Cliente Oracle con ODP.NET

Para el desarrollo de la capa de acceso a datos a una BD Oracle desde .NET es necesario instalar un cliente de Oracle que incluya ODP.NET.

Para este proyecto se ha utilizado el cliente de Oracle “ODAC 12c Release 4 and Oracle Developer Tools for Visual Studio (12.1.0.2.4)”. Descargado de la siguiente URL:

<http://www.oracle.com/technetwork/topics/dotnet/utilsoft-086879.html>

Es importante instalar primero Visual Studio y después el cliente Oracle. De este modo reconocerá que hay una instalación de Visual Studio y se le podrá indicar que se integre con la versión de Visual Studio instalada (debe ser como mínimo Visual Studio 13). De este modo la integración de los componentes ODP.NET se hará sin problemas.

Para poder utilizar ODP.NET en el proyecto habrá que añadir en primer lugar una referencia al ensamblado Oracle.DataAccess. La siguiente ilustración aclara este punto:

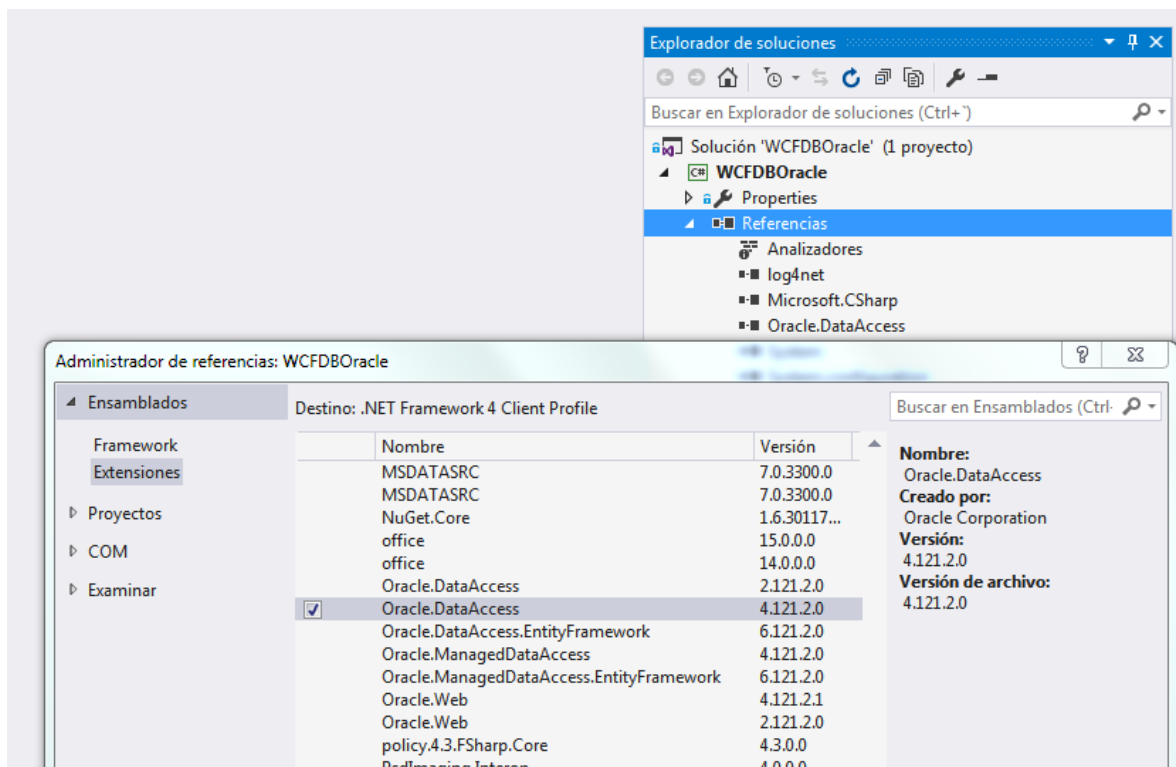


Ilustración 73. Agregar referencia a componente ODP.NET de Oracle.

Y además incluir la siguiente cláusula using para utilizar el siguiente namespace que contiene las clases necesarias para interactuar con BD Oracle:

```
using Oracle.DataAccess.Client;
```

7.3 Log4net

Log4Net es una librería licenciada bajo “Apache Software Licence” que permite incluir en proyectos de Visual Studio la funcionalidad de logging de forma bastante sencilla.

Ofrece distintos niveles de monitorización predefinidos, ordenados de menor prioridad a mayor prioridad:

- Debug
- Info
- Warn
- Error
- Fatal

También se pueden crear niveles personalizados.

Además tenemos distintos tipos de appenders los cuales nos permiten generar logs por ejemplo:

- A través de consola
- En fichero
- Por email
- En BD
- Etc.

Para poder utilizar log4net, dentro de nuestro proyecto, lo primero que tenemos que hacer es instalar el paquete Log4net mediante el administrador de paquetes NuGet.

Una vez instalado hay que crear la referencia al paquete en el fichero packages.config

```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="log4net" version="2.0.4" targetFramework="net40-Client" />
</packages>
```

Dentro del fichero App.Config hay que configurar los loggers. En primer lugar hay que definir una nueva sección (dentro de <configSections>) para log4net:

```
<configSections>
  <section name="log4net" type="log4net.Config.Log4NetConfigurationSectionHandler,
log4net" />
</configSections>
```

A continuación en la sección definida “log4net” se declaran los appender que se van a utilizar dentro de la sección <root> y se define cada uno de ellos en una sección <appender>.

A los appender se les puede aplicar un patrón para formatear la salida y definir cual es el nivel máximo y mínimo de logs generado. Si es un appender para enviar logs por email, en el appender se define la configuración del remitente, destinatario y el servidor de correo. Pero no voy a entrar en más detalles de configuración por que para eso ya existe documentación y tutoriales mucho más completos (al final de este subtítulo se incluyen algunas referencias) mejor mostrar un ejemplo:

```
<log4net debug="false">
  <root>
    <priority value="DEBUG"/>
```

```

    <appender-ref ref="ConsoleAppender"/>
    <appender-ref ref="FileAppender"/>
</root>

<!-- LOGS POR CONSOLA -->
<appender name="ConsoleAppender" type="log4net.Appender.ConsoleAppender">
    <layout type="log4net.Layout.PatternLayout">
        <conversionPattern value="[%level] %date{yyyy-MM-dd HH:mm:ss,fff}
%logger - %message%newline"/>
    </layout>
    <filter type="log4net.Filter.LevelRangeFilter">
        <levelMin value="INFO"/>
        <levelMax value="FATAL"/>
    </filter>
</appender>

<!-- LOGS EN FICHERO -->
<appender name="FileAppender" type="log4net.Appender.FileAppender">
    <file value="C:/Logs/monitorImpresion.log"/>
    <appendToFile value="true"/>
    <layout type="log4net.Layout.PatternLayout">
        <conversionPattern value="[%date{dd-MM-yyyy HH:mm:ss,fff}]
[%level] -%logger- %message %newline"/>
    </layout>
    <filter type="log4net.Filter.LevelRangeFilter">
        <levelMin value="DEBUG"/>
        <levelMax value="FATAL"/>
    </filter>
</appender>

<!-- LOGS POR EMAIL -->
<appender name="SmtpAppenderPrintJob"
type="log4net.Appender.SmtpAppender,log4net">
    <threshold value="WARN"/>
    <to value="destinatario@dominio.es" />
    <from value="remitente@dominio.es"/>
    <subject value="Monitor Impresion - Trabajo de impresion en curso"/>
    <smtpHost value="smtp.dominio.es"/>
    <port value="25"/>
    <authentication value="Basic"/>
    <bufferSize value="1"/>
    <lossy value="true"/>
    <evaluator type="log4net.Core.LevelEvaluator">
        <threshold value="WARN"/>
    </evaluator>
    <filter type="log4net.Filter.LevelRangeFilter">
        <levelMin value="WARN"/>
        <levelMax value="WARN"/>
    </filter>
    <layout type="log4net.Layout.PatternLayout,log4net">
        <conversionPattern value="%property{log4net:HostName} -%logger-
:: %level :: %message
%newlineLogger: %logger%newlineThread: %thread%newlineDate:
%date%newlineNDC: %property{NDC}%newline%newline"/>
    </layout>
</appender>
</log4net>

```

Por último para poder utilizar log4net en una clase hay que referenciar el espacio de nombres con una clausula using:

```
using log4net;
```

Y declarar un objeto de logger para poder utilizarlo dentro de la clase donde se quiere generar logs:

```
private static readonly ILog Log = LogManager.GetLogger("AnalyzeJobId");
```

Para generar logs bastaría utilizar la interface “Log” de la siguiente manera:

```
Log.Debug("Log a nivel de Debug");  
Log.Info("Log a nivel de Info");  
Log.Warn("Log a nivel de Warn");  
Log.Error("Log a nivel de Error");  
Log.Fatal("Log a nivel de Fatal");
```

De esta manera tan sencilla podemos generar logs.

Para profundizar más ver la referencia (Apache Log4Net, n.d.) o el tutorial (Corey, 2015)

7.4 Referencia de archivos del proyecto con código fuente.

Todos los archivos con código fuente están publicados en el repositorio GitHub:

<https://github.com/millarregui/MonitorImpresion/tree/master>

Una vez descargado el zip del repositorio tenemos 2 directorios con 2 proyectos.

El archivo de proyecto para el Servicio Monitor de impresión se encuentra en:

- .\ServicioMonitorImpresion\MonitorImpresion.sln

El archivo de proyecto para el servicio WCF que implementa la capa CAD en:

- .\WCFBDOracle\WCFBDOracle.sln

En ambos casos cuando se carga el proyecto por primera vez hay que cargar la librería log4net. Para ello hay que abrir la consola del administrador de paquetes NuGet y ejecutar el siguiente comando:

```
PM> Install-Package log4net
```

Para la capa CAD también hay que agregar la referencia a ODP.NET de Oracle (debemos tener correctamente instalado el cliente Oracle con ODP.NET con las indicaciones dadas en el anexo). Para ello debemos agregar una referencia a extensión Oracle.DataAccess (dentro de ensamblados -> Extensiones).

7.4.1 Servicio Monitor Impresión

7.4.1.1 Archivo .\ServicioMonitorImpresion\MonitorImpresion\AnalyzeJobId.cs

7.4.1.1.1 Clases

- class **MonitorTrabajosImpresion.AnalyzeJobId**

Con esta clase se obtienen las propiedades de los trabajos de impresión.

Namespaces:

- namespace **MonitorTrabajosImpresion**

7.4.1.2 Archivo .\ServicioMonitorImpresion\MonitorImpresion\ApiImpresion.cs

7.4.1.2.1 Clases

- class **MonitorTrabajosImpresion.ApiImpresion**

Esta clase utiliza la interface de la API del Sistema de impresión para obtener los detalles de los trabajos de impresión.

Namespaces:

- namespace **MonitorTrabajosImpresion**

7.4.1.3 Archivo .\ServicioMonitorImpresion\MonitorImpresion\EmfSpool.cs

7.4.1.3.1 Clases

- class **MonitorTrabajosImpresion.EmfSpool**

Con esta clase se analizan los ficheros de spool con contenido EMF.

Namespaces:

- namespace **MonitorTrabajosImpresion**

7.4.1.4 Archivo .\ServicioMonitorImpresion\MonitorImpresion\EnumDictionaryStruct.cs

7.4.1.4.1 Clases, estructuras, diccionarios y enumeraciones utilizados por las clases principales.

- class **MonitorTrabajosImpresion.ENUM_JPL**
Clase con enumeraciones para analizar cadenas JPL.
- class **MonitorTrabajosImpresion.DICTIONARY_JPL**
Clase con diccionarios para analizar cadenas JPL.
- class **MonitorTrabajosImpresion.ENUM_PCLXL**
Clase contenedora de enumeraciones para analizar contenido PCLXL.
- class **MonitorTrabajosImpresion.DICTIONARY_PCLXL**
Clase con diccionarios para análisis PCLXL
- class **MonitorTrabajosImpresion.STRUCT_PCLXL**
Clase con definición de estructuras para analisis PCLXL.
- class **MonitorTrabajosImpresion.ENUM_API_PRINTING**
Clase contenedora de enumeraciones utilizadas para interactuar con las API del sistema de impresión.
- class **MonitorTrabajosImpresion.STRUCT_API_PRINTING**
Clase contenedora de estructuras utilizadas para interactuar con las API del sistema de impresión.
- class **MonitorTrabajosImpresion.DICTIONARY_API_PRINTING**
Clase con diccionarios utilizados en la clase ApiImpresion juntos a las funciones de la API del Sistema de Impresión.
- class **MonitorTrabajosImpresion.ENUM_EMF**
Clase con enumeraciones utilizadas para análisis EMF.
- class **MonitorTrabajosImpresion.DICTIONARY_EMF**
Clase con diccionarios utilizados para análisis EMF.
- class **MonitorTrabajosImpresion.ENUM_SYSTEM_PRINTING**
Clase con enumeraciones utilizadas en la clase LocalPrinting.
- struct **MonitorTrabajosImpresion.STRUCT_PRINT_JOB**
Struct con las propiedades que recopilamos para cada trabajo de impresión.
- class **MonitorTrabajosImpresion.STRUCT_WATCHIO**
Clase con estructuras para interactuar con el sistema de Archivos de Windows.
- class **MonitorTrabajosImpresion.Tags**
Clase para almacenar items. La utilizamos para guardar un resumen de los distintos Items (o tags) encontrados en el análisis de los ficheros de Spool.

Namespaces:

- namespace **MonitorTrabajosImpresion**

7.4.1.5 Archivo .\ServicioMonitorImpresion\MonitorImpresion\EventMonitorPrinting.cs

7.4.1.5.1 Clases

- class **MonitorTrabajosImpresion.EventMonitorPrinting**

Clase para monitorizar los eventos del Sistema de Impresión.

Namespaces:

- namespace **MonitorTrabajosImpresion**

7.4.1.6 Archivo .\ServicioMonitorImpresion\MonitorImpresion\JPLSpool.cs

7.4.1.6.1 Clases

- class **MonitorTrabajosImpresion.JPLSpool**

Clase para analizar ficheros de spool con contenido JPL.

Namespaces:

- namespace **MonitorTrabajosImpresion**

7.4.1.7 Archivo .\ServicioMonitorImpresion\MonitorImpresion\LocalPrinting.cs

7.4.1.7.1 Clases

- class **MonitorTrabajosImpresion.LocalPrinting**

Permite obtener propiedades de Impresión utilizando clases del namespace System.Printing.

Namespaces:

- namespace **MonitorTrabajosImpresion**

7.4.1.8 Archivo .\ServicioMonitorImpresion\MonitorImpresion\PCLXLSpool.cs

7.4.1.8.1 Clases

- class **MonitorTrabajosImpresion.PCLXLSpool**

Clase que implementa el análisis de ficheros de **Spool** con contenido PCLXL

- struct **MonitorTrabajosImpresion.PCLXLSpool.PositionTagPCLXL**

Estructura para guardar la posición de los Tag PCLXL que se van analizando.

Namespaces:

- namespace **MonitorTrabajosImpresion**

7.4.1.9 Archivo .\ServicioMonitorImpresion\MonitorImpresion\Program.cs

7.4.1.9.1 Clases

- class **MonitorTrabajosImpresion.Programa**

Clase que sirve de punto de entrada al programa. Contiene el método main().

Namespaces:

- namespace **MonitorTrabajosImpresion**

7.4.1.10 Archivo .\ServicioMonitorImpresion\MonitorImpresion\ProjectInstaller.cs

7.4.1.10.1 Clases

- class **MonitorImpresion.ProjectInstaller**

Clase que forma parte del instalador creado para poder instalar el Monitor de Impresión como Servicio del Sistema.

Namespaces:

- namespace **MonitorImpresion**

7.4.1.11 Archivo .\ServicioMonitorImpresion\MonitorImpresion\ProjectInstaller.Designer.cs

7.4.1.11.1 Clases

- class **MonitorImpresion.ProjectInstaller**

Clase que forma parte del instalador creado para poder instalar el Monitor de Impresión como Servicio del Sistema.

Namespaces:

- namespace **MonitorImpresion**

7.4.1.12 Archivo .\ServicioMonitorImpresion\MonitorImpresion\ServicePrintMonitor.cs

7.4.1.12.1 Clases

- class **MonitorTrabajosImpresion.ServicePrintMonitor**

clase para iniciar el monitor de impresión como servicio de sistema.

Namespaces:

- namespace **MonitorTrabajosImpresion**

7.4.1.13 Archivo .\ServicioMonitorImpresion\MonitorImpresion\Spool.cs

7.4.1.13.1 Clases

- class **MonitorTrabajosImpresion.Spool**

Clase base para análisis de ficheros de **Spool**.

Namespaces:

- namespace **MonitorTrabajosImpresion**

7.4.1.14 Archivo .\ServicioMonitorImpresion\MonitorImpresion\Util.cs

7.4.1.14.1 Clases

- class **MonitorTrabajosImpresion.Util**
Clase con funciones de uso general.
- class **MonitorTrabajosImpresion.ElapsedTime**
Clase que permite controlar un tiempo de espera.

Namespaces:

- namespace **MonitorTrabajosImpresion**

7.4.1.15 Archivo .\ServicioMonitorImpresion\MonitorImpresion\WatchIO.cs

7.4.1.15.1 Clases

- class **MonitorTrabajosImpresion.WatchIO**
Permite interactuar con el Sistema de Archivos y realizar backup de los ficheros de **Spool**.

Namespaces

- namespace **MonitorTrabajosImpresion**

7.4.1.16 Archivo .\ServicioMonitorImpresion \MonitorImpresionTest\Test.cs

7.4.1.16.1 Clases

- class **MonitorImpresionTest.EvenMonitorPrintingTest**
Pruebas sobre la clase EvenMonitorPrinting.
- class **MonitorImpresionTest.UtilTest**
Pruebas sobre la clase Util.
- class **MonitorImpresionTest.EmfSpoolTest**
Pruebas sobre la clase EmfSpool.
- class **MonitorImpresionTest.JPLSpoolTest**
Pruebas sobre la clase JPLSpool.
- class **MonitorImpresionTest.TagsTest**
Pruebas sobre la clase Tags.
- class **MonitorImpresionTest.ApiImpresionTest**
Pruebas sobre la clase ApiImpresion.
- class **MonitorImpresionTest.WatchIOTest**
Pruebas sobre la clase WatchIO.

Namespaces:

- namespace **MonitorImpresionTest**

7.4.2 Servicio WCF para implementar la capa CAD

7.4.2.1 Archivo `.\WCFDBOracle\WCFDBOracle\DB.cs`

7.4.2.1.1 Clases

- class **WCFDB.DB**

Clase que implementa la interface IDB del servicio WCF. Se sirve de la clase DBOracle para interactuar con la BD donde se guardan las propiedades de los trabajos de impresión.

Namespaces:

- namespace **WCFDB**

7.4.2.2 Archivo `.\WCFDBOracle\WCFDBOracle\DBOracle.cs`

7.4.2.2.1 Clases

- class **WCFDB.DBOracle**

Clase de Conexión y empleo de la **DB** Oracle. ODP.NET Oracle managed provider.

Namespaces:

- namespace **WCFDB**

7.4.2.3 Archivo `.\WCFDBOracle\WCFDBOracle\IDB.cs`

7.4.2.3.1 Clases

- interface **WCFDB.IDB**

Interface del Servicio WCF que implementa la capa CAD, donde se publican los métodos que serán accesibles desde el Servicio Monitor de Impresión.

Namespaces:

- namespace **WCFDB**

8. GLOSARIO DE TÉRMINOS

API: de Wikipedia: *“La interfaz de programación de aplicaciones, abreviada como API (del inglés: Application Programming Interface), es el conjunto de subrutinas, funciones y procedimientos(o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizada por otro software como una capa de abstracción.”.*

b/n, B/N: Blanco y negro.

BD: Base de datos.

Big Endian: designa el formato en el que se almacenan los datos de más de un byte. Consiste en representar los bytes internamente en su orden natural es decir de la misma manera que se escribe, los bytes más significativos primero.

CAD: Capa de Acceso a Datos.

EMF: Enhanced MetaFile.

EMR: Enhanced Metafile Records.

GDI: Graphic Device Interface

GUI: Grafical User Interface.

HW: Hardware.

IDE: Integrated User Interface: Ambiente de desarrollo Integrado.

JPL: Printer Job Language.

LDP: Lenguaje de descripción de página (del termino inglés PDL).

Little Endian: designa el formato en el que se almacenan los datos de más de un byte. Los bytes no se representan internamente en su orden natural, es decir de la misma manera que se escribe, sino los menos significativos primero.

MSDN: Microsoft Developer Network. Extraído de Wikipedia:

“Tiene como objeto la resolución de dudas y problemas que le puedan surgir al desarrollador; según la propia corporación “Contiene una gran cantidad de información técnica de programación, incluidos código de ejemplo, documentación, artículos técnicos y guías de referencia”, en este último caso la denominación correcta sería “API o biblioteca MSDN”, de la cual también se encuentra una versión en línea.”.

ODP.NET: Oracle Data Provider for .NET.

PAS: Personal de Administración y servicios.

PDL: Page description language.

PCLXL: HP Page description Language Level 6.

PDI: Personal Docente e Investigador.

POO: Programación orientada a objetos.

S.G.B.D: Sistema de Gestión de Base de Datos.

S.O: Sistema Operativo.

STIC: Servicio de las Tecnologías de la Información y las Comunicaciones.

SW: Software.

TFG: Trabajo Fin de Grado.

UAL: Universidad de Almería.

UEL Command: (the Universal Exit Language (UEL) command): cadena "<ESC>%-12345X"

WCF (Windows Communication Foundation): Conjunto de librerías que provee Microsoft en .NET para construir aplicaciones orientadas a ofrecer servicios a través de internet, estos pueden estar hospedados en un servidor IIS o en una aplicación.

WPF (*Windows Presentation Foundation*): De Wikipedia *"es una tecnología de Microsoft, presentada como parte de Windows Vista. Permite el desarrollo de interfaces de interacción en Windows tomando características de aplicaciones Windows y de aplicaciones web.*

WPF ofrece una amplia infraestructura y potencia gráfica con la que es posible desarrollar aplicaciones visualmente atractivas, con facilidades de interacción que incluyen animación, vídeo, audio, documentos, navegación o gráficos 3D. Separa, con el lenguaje declarativo XAML y los lenguajes de programación de .NET, la interfaz de interacción de la lógica del negocio, propiciando una arquitectura Modelo Vista Controlador para el desarrollo de las aplicaciones.

WPF se incluye con Windows 7, Windows Vista y Windows Server 2008 y también está disponible para Windows XP Service Pack 2 o posterior y Windows Server 2003."

XPS (XML Paper Specification): Es un formato de documento electrónico que utiliza como base XML. Referido al Sistema de impresión, se puede utilizar como lenguaje PDL para guardar trabajos de impresión en ficheros de spool.

RESUMEN

En este documento se explica el funcionamiento del sistema de impresión de Windows y su interacción con otros componentes del sistema operativo. El Sistema de impresión de Windows registra su actividad en el registro de eventos de Windows (eventos de impresión) y almacena los trabajos de impresión en ficheros de spool empleando alguno de los lenguajes de descripción de páginas (PDL). El objetivo de este TFG es poder obtener las propiedades de los trabajos de impresión con las que poder calcular sus costes de impresión. A través de los eventos de impresión es posible identificar los trabajos de impresión (Número de trabajo de impresión, impresora, Servidor de impresión, usuario propietario). Mediante el namespace System.Printing de .NET y las APIs del sistema de impresión se obtienen propiedades como: nombre del documento, número de páginas, impresión en color o b/n, impresión simple o doble cara, tamaño de página, fecha de impresión. Analizando los ficheros de spool (empleando lenguajes PDL: EMFSPPOOL/EMF o JPL/PCLXL), es posible obtener ciertas propiedades como el número de páginas con contenido en color o el tamaño de cada una de las páginas del documento. Se utiliza Visual Studio (con las extensiones log4net y ODP.NET) y C# para crear un Servicio Windows con el que monitorizar los trabajos de impresión y un servicio WCF para crear la capa de acceso a datos para almacenar las propiedades de los trabajos de impresión en una BD Oracle. Destacar que surge como necesidad utilizar un marco de trabajo como SCRUM, junto Visual Studio TFS, para conducir la última fase de este TFG.

ABSTRACT

This document shows the operation of Windows printing system and its interaction with other components of the operating system. The Windows printing system allows to record its activity in the Windows even log (printing events) and it stores printjobs in the spool files by using any of the page description languages (PDL). The aim of this project is to obtain the properties of print jobs in order to calculate their printing costs. The identification of printing jobs is possible by means of the printing events (such as number of print job, printer, print server, owner user). Different features, like document name, number of pages, print in color or b/n, simple or double-sided printing, page size, print date, among others, can be obtained using the .NET namespace System.Printing and the API printing system. Moreover, analyzing spool files (using PDL languages as EMFSPPOOL/EMF or JPL/PCLXL), it is possible to obtain certain properties such as number of pages with color content or the size of each page of the document. Finally, let us remark that Visual Studio (with log4net and Orade ODP.NET extensions) and C# are used in order to create a Windows and WFC services, which allows to monitor print jobs and to create de data access layer in order to save the properties of printjobs in an Oracle database, respectively. In order to complete the last stage of this project, it is important to remark the use of a developer framework such as Scrum with Visual Studio Team Foundation Server.

